Interim Report on Fuel Cycle Neutronics Code Development in FY2008

Global Nuclear Energy Partnership

Prepared for
U.S. Department of Energy
Reactor Campaign

Cristian Rabiti, Micheal A Smith, Dinesh Kaushik, and Won Sik Yang

March 31, 2008
ANL-AFCI-222

Prepared by:

Principle Investigator — 3-31-08
Cristian Rabiti — Date

Principle Investigator — 3-31-08
Micheal A. Smith — Date

Principle Investigator — 3/31/08
Dinesh Kaushik — Date

Reviewed by:

GNEP Work Package Manager — 3-31-08
Won Sik Yang — Date

GNEP WBS Level 4 Manager — 3-31-08
Andrew R. Siegel — Date

Approved by :

GNEP Reactor Campaign Director — 3-27-08
Robert N. Hill — Date

SUMMARY

As part of the Global Nuclear Energy Partnership (GNEP), a fast reactor simulation program was launched in April 2007 to develop a suite of modern simulation tools specifically for the analysis and design of sodium cooled fast reactors. The general goal of the new suite of codes is to reduce the uncertainties and biases in the various areas of reactor design activities by enhanced prediction capabilities. Under this fast reactor simulation program, a high-fidelity deterministic neutron transport code named UNIC is being developed. The final objective is to produce an integrated, advanced neutronics code that allows the high fidelity description of a nuclear reactor and simplifies the multi-step design process by direct coupling with thermal-hydraulics and structural mechanics calculations.

Currently there are three solvers for the neutron transport code incorporated in UNIC: PN2ND, SN2ND, and MOCFE. PN2ND is based on a second-order even-parity spherical harmonics discretization of the transport equation and its primary target area of use is the existing homogenization approaches that are prevalent in reactor physics. MOCFE is based upon the method of characteristics applied to an unstructured finite element mesh and its primary target area of use is the fine grained nature of the explicit geometrical problems which is the long term goal of this project. SN2ND is based on a second-order, even-parity discrete ordinates discretization of the transport equation and its primary target area is the modeling transition region between the PN2ND and MOCFE solvers.

The major development goal in fiscal year 2008 for the MOCFE solver was to include a two-dimensional capability that is scalable to hundreds of processors. The short term goal of this solver is to solve two-dimensional representations of reactor systems such that the energy and spatial self-shielding are accounted for and reliable cross sections can be generated for the homogeneous calculations. In this report we present good results for an OECD benchmark obtained using the new two-dimensional capability of the MOCFE solver. Additional work on the MOCFE solver is focused on studying the current parallelization algorithms that can be applied to both the two- and three-dimensional implementations such that they are scalable to thousands of processors. The initial research into this topic indicates that, as expected, the current parallelization scheme is not sufficiently scalable for the detailed reactor geometry that it is intended for. As a consequence, we are starting the investigative research to determine the alternatives that are applicable for massively parallel machines.

The major development goal in fiscal year 2008 for the PN2ND and SN2ND solvers was to introduce parallelism by angle and energy. The motivation for this is two-fold: 1) reduce the memory burden by picking a simpler preconditioner with reduced matrix storage and 2) improve parallel performance by solving the angular subsystems of the within group equation simultaneously. The solver development in FY2007 focused on using PETSc to solve the within group equation where only spatial parallelization was utilized. Because most homogenous problems required relatively few spatial degrees of freedom (tens of thousands) the only way to improve the parallelism was to spread the angular moment subsystems across the parallel system. While the coding has been put into place for parallelization by space, angle, and group, we have not optimized any of the solvers and therefore do not give an assessment of the achievement of this work in this report. The immediate task to be completed is to implement and validate Tchebychev acceleration of the fission source iteration algorithm (inverse power method in this work) and optimize both the PN2ND and SN2ND solvers. We further intend to extend the applicability of the UNIC code by adding a first-order discrete ordinates solver termed SN1ST.

Upon completion of this work, all memory usage problems are to be identified and studied in the solvers with the intent of making the new version of an exportable production code in either FY2008 or FY2009. This report covers the status of these tasks and discusses the work yet to be completed.

CONTENTS

FIGURES

TABLES

REACTOR CAMPAIGN
INTERIM REPORT ON FUEL CYCLE NEUTRONICS CODE DEVELOPMENT IN FY2008

1. Introduction

As part of the Global Nuclear Energy Partnership (GNEP), a fast reactor simulation program was launched in April 2007 to develop a suite of modern simulation tools specifically for the analysis and design of sodium cooled fast reactors. The general goal of the new suite of codes is to reduce the uncertainties and biases in the various areas of reactor design activities by enhanced prediction capabilities. Under this fast reactor simulation program, a high-fidelity deterministic neutron transport code named UNIC is being developed [1]. The final objective is to produce an integrated, advanced neutronics code that allows the high fidelity description of a nuclear reactor and simplifies the multi-step design process by direct coupling with thermal-hydraulics and structural mechanics calculations.

In FY2007, we formally introduced three solvers for the neutron transport code which where termed: PN2ND, SN2ND, and MOCFE. PN2ND is based on a second-order even-parity spherical harmonics discretization of the transport equation and its primary target area of use is the existing homogenization approaches that are prevalent in reactor physics [2]; however, we note that additional supporting algorithms are necessary to fully handle the wide range of homogenization methods. MOCFE is based upon the method of characteristics applied to an unstructured finite element mesh and its primary target area of use is the fine grained nature of the explicit geometrical problems which is the long term goal of this project [3]. SN2ND is based on a second-order, even-parity discrete ordinates discretization of the transport equation and its primary target area is the modeling transition region between the PN2ND and MOCFE solvers [2].

There were major development goals for all three solvers specified for FY2008. Even with current state of the art computer technology, we cannot realistically perform explicit geometry calculations with sufficient energy resolution to guarantee accuracy of the modeling. Consequently, the current development focus is to finalize the solvers that target the homogenization methodologies and begin developing the necessary design analysis codes for fuel cycle analysis, transient analysis and perturbation theory calculation using them.

To support this work, the first goal is to develop a two-dimensional MOCFE capability which is scalable to hundreds of processors. The short term goal of this solver is to solve two-dimensional representations of reactor systems such that the energy and spatial self-shielding are accounted for and reliable cross sections can be generated for the homogeneous calculations. While not tasked for this year, this requires the development of a heterogeneous to homogeneous mesh mapping algorithm that will take the solution from the MOCFE solver and produce cross sections for the homogeneous problem. Additional work on the MOCFE solver is focused on studying the current parallelization algorithms that can be applied to both the two- and three-dimensional implementations such that they are scalable to thousands of processors.

In addition to work on the MOCFE solver, research on improving the parallelism of the PN2ND and SN2ND solvers was undertaken. As discussed previously, these solvers rely heavily upon the PETSc toolkit for linear algebra in a parallel environment. The initial solver development focused on using PETSc to solve the within group equation where only spatial parallelization was utilized. The new development focus is to reduce the memory burden and improve parallelization by introducing parallelization by angle and energy. Tchebychev acceleration of the fission source iteration algorithm (inverse power method in this work) is also to be implemented and validated.

Upon completion of this work, all memory usage problems are to be identified and studied in the solvers with the intent of making the new version of an exportable production code in either FY2008 or FY2009. In this report we discuss the current status of these tasks and the work left to be completed. Given the presence of reliable and accurate solvers, the additional development focus for FY2008 is to produce a first version of kinetics analysis tool. This work along with some detailed benchmarking calculations and creation of a first-order discrete ordinates discretization of the transport equation, termed SN1ST, is scheduled to start in the second part of FY2008 and thus is not discussed in this report.

2. Method of Characteristics Solver MOCFE Developments

The method of characteristics (MOC) poses an advantage over the other methods incorporated in the UNIC code (PN2ND, SN2ND, and SN1ST) for solving the transport equation due to its ability to cheaply handle and solve fine mesh representations of the fuel pin geometry. This capability comes at the expense of a very low order spatial approximation of the source (flat) within each element. As one can expect, a flat source approximation strongly affects the ability of the method to handle large elements which are more than a neutron mean free path in size. In general, this makes the MOCFE solver disadvantageous when applied to large homogenized problems and thus the other solvers in UNIC are more appropriate. However, when the pin by pin description of the geometry is defined with ring wise depletion zones in each pin, the other solvers in UNIC are very expensive with regard to both memory and computational effort and thus the MOCFE solver is a much more desirable solver.

As it turns out, the ability of MOC (and by extension the collision probability method) to handle the fine granularity mesh sizes has led it to be widely applied in lattice cell calculations from which homogenized cross sections are typically obtained [2]. These homogenization processes assume that the flux solution derived from a lattice cell calculation is sufficiently representative of the flux solution in the full reactor system that the flux solution for a reactor can be constructed without solving the explicit geometrical representation of the entire core. The lattice cell calculations are typically two-dimensional calculations of a repeated geometry structure in the reactor such as a single fuel pin or an entire fuel assembly. Reflected boundary conditions are imposed upon the surface of the lattice cell and a critical buckling search is performed to approximate the conditions that exist for each assembly in the real reactor system (rather than the lattice cell). As one would expect, this approach becomes less reliable as the heterogeneity of the core increases and the gradient in the flux between adjacent assemblies (or fuel pins) becomes steeper.

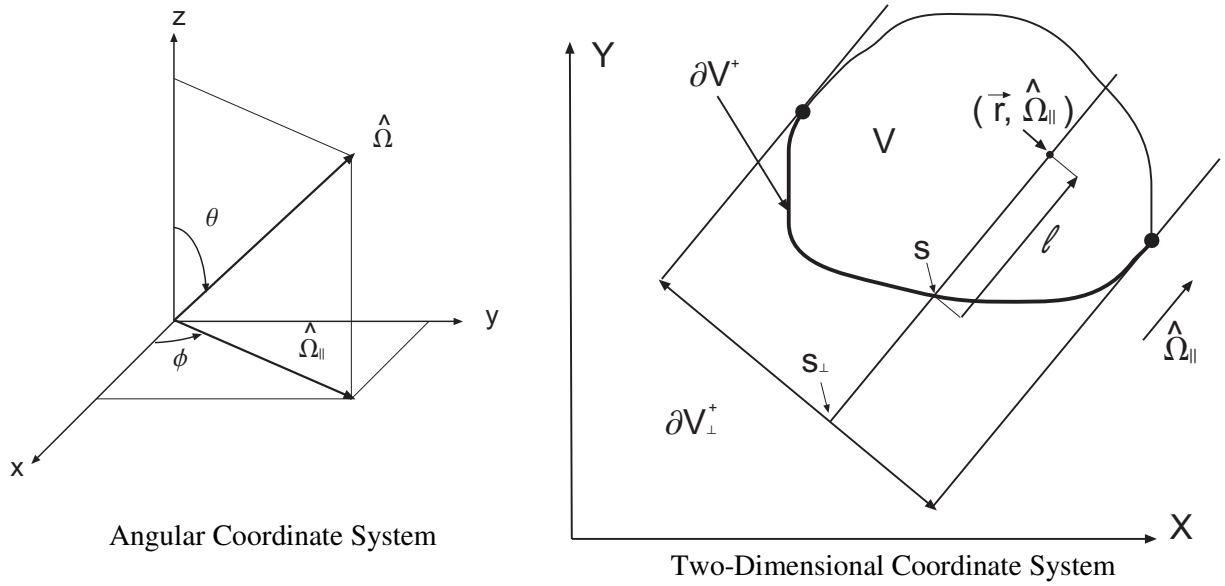
While explicit geometry three-dimensional calculations are still beyond the abilities of modern computing technology, it is becoming evident that explicit geometry two-dimensional planar calculations are not. Consequently, the development of a two-dimensional MOCFE solver that is massively parallel not only provides the ability to handle the existing homogenization approaches, but also the next logical step in homogenization treatments which is to utilize the flux solution for an entire radial plane of the reactor system as a means to generate homogeneous coefficients for the three-dimensional geometry. With such an approach, the issues that exist for current homogenization methods are likely to be reduced if not eliminated entirely.

2.1 Derivation of the Two-Dimensional MOC

To begin the two-dimensional method of characteristics formulation, we define the reference systems shown in Figure I. Similar to the three-dimensional derivation, we again need to define an area $s_{\perp}$ on a plane external to the domain V, that is perpendicular to the direction of neutron travel $\hat{\Omega}$. In two-dimensions, we project $\hat{\Omega}$ onto the x-y plane to define $\hat{\Omega}_{\parallel}$. Given that we can respectively identify the incident and exiting surfaces of the domain V as ∂V^{+} and ∂V^{-}, we can pick a plane P external to the

domain V and orthogonal to $\hat{\Omega}$, which for two-dimensional geometries is a line in the x-y plane, and project the incident portion of the surface onto this plane to get $\partial V_{\perp}^{+}$ as shown in Figure I. For two-dimensional geometries we can write this as Eqs. (2.1).

$$\begin{aligned} \partial V^{+} &= \left\{ \partial V \mid \left(\cos\phi\sin\theta, \sin\phi\sin\theta, 0\right) \cdot \left(n_x, n_y, 0\right) > 0 \right\} \\ \partial V_{\perp}^{+} &= \text{Projection of } \partial V^{+} \text{ onto } P \text{ in parallel to } \left(\cos\phi\sin\theta, \sin\phi\sin\theta, 0\right) \end{aligned} \tag{2.1}$$



Angular Coordinate System

Two-Dimensional Coordinate System

Figure I. Two-Dimensional MOC Coordinate System

This makes $s_{\perp}$ a one-dimensional coordinate on the projected perimeter $\partial V_{\perp}^{+}$. The next step in the ray tracing procedure is to define ℓ as the distance between a point s on ∂V^{+} and some position $\vec{r} \in V$ within the volume along the direction $\hat{\Omega}$. Once again, in two-dimensional geometry we can impose $\vec{r} = \left(x, y\right)$ and look for the distance ℓ along the projected direction $\hat{\Omega}_{\parallel}$ as shown in Figure I. If the problem domain is convex, then the position $\left(s_{\perp}, \ell, \hat{\Omega}\right)$ defines an alternate coordinate system to $\left(x, y, \hat{\Omega}\right)$ and the within group neutron transport equation can be written as

$$\left(\frac{\partial}{\partial \ell / \sin\theta} + \Sigma_t\left(s_{\perp}, \ell\right) \right) \varphi\left(s_{\perp}, \ell, \hat{\Omega}\right) = \int_{4\pi} \Sigma_s\left(s_{\perp}, \ell, \hat{\Omega} \cdot \hat{\Omega}'\right) \varphi\left(s_{\perp}, \ell, \hat{\Omega}'\right) d\hat{\Omega}' + S\left(s_{\perp}, \ell, \hat{\Omega}\right) \tag{2.2}$$

where the group index has been suppressed. For every different value of $\hat{\Omega}$, a different spatial coordinate system $\left(s_{\perp}, \ell\right)$ can be defined, however, since θ is bounded to satisfy $0 < \theta < \pi/2$, the construction of the coordinate system $\left(s_{\perp}, \ell, \hat{\Omega}\right)$ is invariant with respect to the value of θ. This last fact permits the use of a single reference system for all $\hat{\Omega}$ having the same ϕ value.

Under the assumption that the cross sections are constant, source and self scattering are constant and homogenous in angle within each 2-D mesh element (V_i) we obtain:

$$\left(\frac{\partial}{\partial \ell / \sin\theta} + \Sigma_t\left(\vec{s}_\perp, \ell\right)\right)\varphi\left(\vec{s}_\perp, \ell, \hat{\Omega}\right) = \Sigma_s^i \frac{1}{V_i}\int_{V_i} d\vec{r} \int_{4\pi} d\hat{\Omega}' \psi_i\left(\vec{r}, \hat{\Omega}'\right) + S_i = Q_i \tag{2.3}$$

Similar to the formulation for three dimensional MOC, the right side of Eq. (2.3) is assumed to be known and an analytic solution to equation (2.3) can be written for each trajectory $t\left(s_\perp, \phi\right)$ that crosses an element. To simplify the notation, we define $R_{i,t}$ as the length of the intersection for a given trajectory $t\left(s_\perp, \phi\right)$ that intersects element V_i. If we view this as a local reference system for the variable ℓ inside of the element V_i, the point $\left(s_\perp, 0, \phi\right)$ becomes the incoming intersection point and $\left(s_\perp, R_i, \phi\right)$ becomes the outgoing intersection point for the trajectory $t\left(s_\perp, \phi\right)$ that intersects element V_i. We can further define $\varphi\left(s_\perp, 0, \hat{\Omega}\right)$ as the incoming angular flux, $\varphi_{i,in}\left(s_\perp, \hat{\Omega}\right)$, at the intersection point $\left(s_\perp, 0, \phi\right)$, and $\varphi\left(s_\perp, R_i, \hat{\Omega}\right)$ as the outgoing angular flux, $\varphi_{i,out}\left(s_\perp, \hat{\Omega}\right)$, for the intersection point $\left(s_\perp, R_i, \phi\right)$. With these definitions, we can write Eq. (2.3) such that we define the equation

$$\varphi_{i,out}\left(s_\perp, \hat{\Omega}\right) = \varphi_{i,in}\left(s_\perp, \hat{\Omega}\right)\exp\left(-\Sigma_{t,i} R_{i,t} / \sin\theta\right) + \frac{1 - \exp\left(-\Sigma_{t,i} R_{i,t} / \sin\theta\right)}{\Sigma_{t,i}} Q_i \ . \tag{2.4}$$

For two adjacent elements *k* and *i* that share the intersection point $\left(s_\perp, R_i, \phi\right)$ for the trajectory $t\left(s_\perp, \phi\right)$, we can write the continuity condition for the flux such that

$$\varphi_{k,in}\left(s_\perp, \hat{\Omega}\right) = \varphi_{i,out}\left(s_\perp, \hat{\Omega}\right). \tag{2.5}$$

We note that the same type of relationship exists for the other intersection point $\left(s_\perp, 0, \phi\right)$ but that the relative nature of the element indices must switch in Eq. (2.5). Given the nature of the solution process – each element is connected to the preceding element along the path of the trajectory to the incident domain boundary – these equations are termed the "propagation" equation.

The propagation equation can be used for all trajectories emanating from the set of points $\left(s_\perp, \phi\right)$ on the incident domain boundary thereby providing a means to propagate the incident angular flux through the domain to the exiting surface of the problem domain. The solution of this system of equations of course provides the neutron angular flux at all of the intersection points for all of the trajectories on all of the elements in the domain. We can project this representation of the angular flux such that the average reaction rate quantities can be constructed within each element. We start by noticing that

$$\frac{1}{V_i}\int_{V_i} d\vec{r} \int_{4\pi} d\hat{\Omega}\, \psi_i\left(\vec{r}, \hat{\Omega}\right) = \frac{1}{V_i}\int_{4\pi} d\hat{\Omega} \int_{\partial V_\perp^+} ds_\perp \int_0^{R_{i,t}} d\ell\, \psi_i\left(s_\perp, \ell, \hat{\Omega}\right) \tag{2.6}$$

for every element in the domain. We can use the analytical solution from Eq. (2.4) to define

$$\Sigma_{t,i}\int_0^{R_{i,t}} d\ell\, \psi_i\left(s_\perp, \ell, \hat{\Omega}\right) = \left(\varphi_{i,in}\left(s_\perp, \hat{\Omega}\right) - \varphi_{i,out}\left(s_\perp, \hat{\Omega}\right)\right)\sin\theta + R_{i,t} Q_i \ . \tag{2.7}$$

We can approximate the integral over $\hat{\Omega}$ and the integral over $s_\perp$ by the implicit sums to define

$$\frac{1}{V_i}\int_{V_i} d\vec{r}\int_{4\pi} d\hat{\Omega}\,\psi_i\left(\vec{r},\hat{\Omega}\right)$$

$$=\frac{1}{V_i}\sum_n w_n\sum_m w_m \sum_{j\ t\left(s_{\perp,j},\hat{\Omega}_{n,m}\right)\cap\partial V_{i,\perp}^{+}} \omega_{\perp,j}\int_0^{R_{i,n,j}} d\ell\,\varphi_{n,m,j}\left(\ell\right) \tag{2.8}$$

$$=\frac{1}{\Sigma_{t,i}V_i}\sum_n w_n\sum_m w_m \sum_{j\ t\left(s_{\perp,j},\hat{\Omega}_{n,m}\right)\cap\partial V_{i,\perp}^{+}} \omega_{\perp,j}\left[\left(\varphi_{i,in}\left(s_\perp,\hat{\Omega}_{n,m}\right)-\varphi_{i,out}\left(s_\perp,\hat{\Omega}_{n,m}\right)\right)\sin\theta_m+R_{i,t}Q_i\right]$$

The weights w_n and w_m for the angular space are typically obtained by utilizing a product cubature where two one-dimensional quadrature in θ and ϕ are combined. The perpendicular surface $s_\perp$ is also broken into segments where again a cubature can be imposed such that the points form the starting points of trajectories and the weights sum to the area of the surface ∂V^+ projected on to the perpendicular plane. The grid points of this cubature $\left(\theta_n,\phi_m,s_{\perp,j}\right)$ define the trajectories for which the preceding equations are solved on. As the propagation equation is solved, it inherently provides contribution from each trajectory to Eq. (2.8). Given the presence of scattering, and in particular within group scattering, Eqs. (2.4), (2.5), and (2.8) can be solved in an iterative manner similar to that the three-dimensional MOCFE solver. As one would expect, the treatment of the boundary conditions, the construction of the Algebraic Collapsing Acceleration scheme for the within group scattering and the multi-group strategy itself are equivalent to the one used in the three-dimensional MOCFE solver and thus is not repeated here.

2.2 Two-Dimensional Ray Tracing

As mentioned in Section 2.1, the construction of the coordinate system $\left(s_\perp,\ell\right)$ is invariant with respect to the value of the angle θ. As a consequence, the ray tracing process only needs to be performed for the unique values of ϕ. As such, we have implemented a product cubature approach for the two-dimensional MOCFE solver such that the Tchebychev quadrature is always used in the ϕ space. This quadrature is ideal because the angles are equally spaced, the weights are equal, and, when combined with the Legendre quadrature, can exactly integrate the spherical harmonics used to represent the scattering kernel.

Figure II shows the current element types that are included in the MOCFE solver. The first part of the ray tracing process is the definition of the starting points for each direction ϕ_m derived from the projected incident surface $\partial V_{\perp,m}^{+}$. In the current approach, all the vertices of the finite element lying on ∂V^+ are individually projected on the segment representing $\partial V_{\perp,m}^{+}$. The number of trajectories $s_{\perp,j}$ between two consecutive points on the segment $\partial V_{\perp,m}^{+}$ is chosen such that the weight $\omega_{\perp,j}$ (equivalent to area) of $\partial V_{\perp,m}^{+}$ assigned to any trajectory is less than a user defined input value. Once the coordinate of the points $s_{\perp,j}$ are fixed, their projection, with direction $\hat{\Omega}_{\parallel}$ on ∂V^+, locates the points s_j that are the incoming point of the characteristics line in the domain. Given the incoming location of a trajectory for a finite element lying on the incident boundary of the domain, a line-segment intersection algorithm is used to find the outgoing point and the intersection length. For quadratic finite elements the curvilinear perimeter is approximated by a tessellation of the vertices as shown in Figure III. This approximation appears

reasonable since the area of any given element within the domain is rarely intersected exactly by a trajectory crossing the domain and a volume rebalance must be imposed. Given the outgoing point and thus outgoing surface of the element is identified, a simple surface to surface element connectivity list provides the next element that the trajectory intersects. This process is repeated until the trajectory reaches the outgoing surface of the domain.

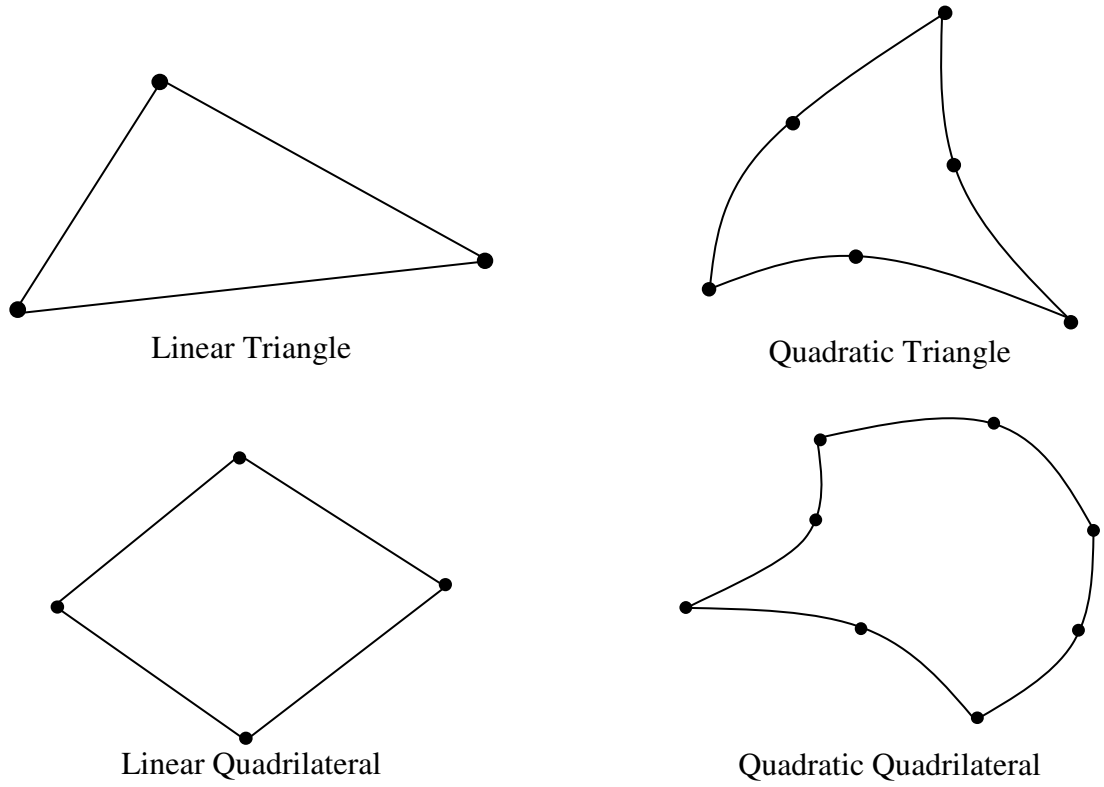


Figure II. Two-dimensional Finite Elements Included in the MOCFE Solver

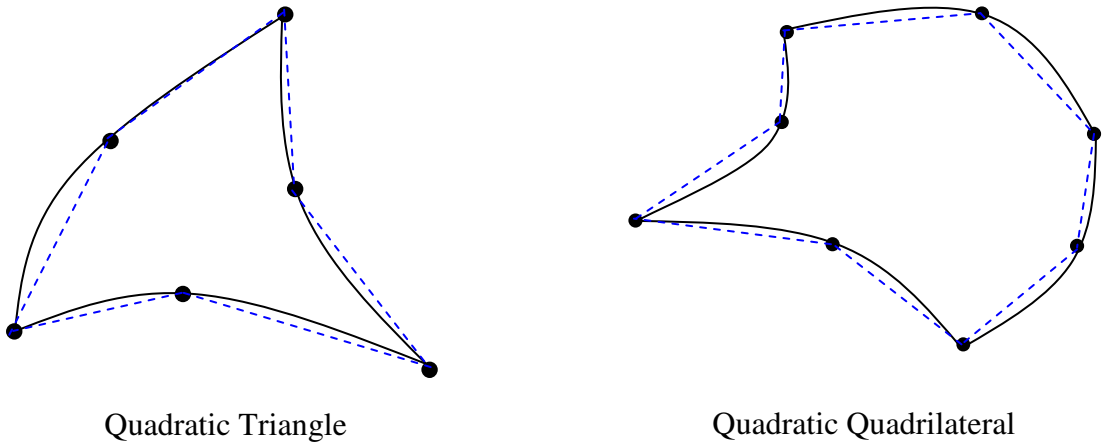


Figure III. Linear interpolation of the Quadratic Finite Elements.

2.3 The C5G7 2-Dimensional Benchmark

The two-dimensional option of the MOCFE solver was tested on numerous benchmark problems to ensure reliability. One such benchmark problem which we used to assess the accuracy of the solver is the OECD C5G7 benchmark [4]. The five finite element meshes in Figure IV show the mesh layout used for each C5G7 pin-cell. To estimate the number of elements necessary for modeling the full geometry, a single UO_2 pin-cell was solved using these meshes with a ray spacing of 0.001 cm, 32 azimuthal directions and 3 polar directions. Table 1 gives the eigenvalue results obtained with the MOCFE solver where the reference MCNP solution was 1.32557±0.00054. As can be seen, there is a relatively large change in the eigenvalue after the 84(b) mesh. This change corresponds with an introduction of more mesh segments in the radial plane (or more segments in the azimuthal angle) as seen in Figure IV. The flux solution in Figure V also indicates the importance of the mesh layout where the coarser azimuthal meshing constrains the distribution of the flux to have more symmetry than is physically present.

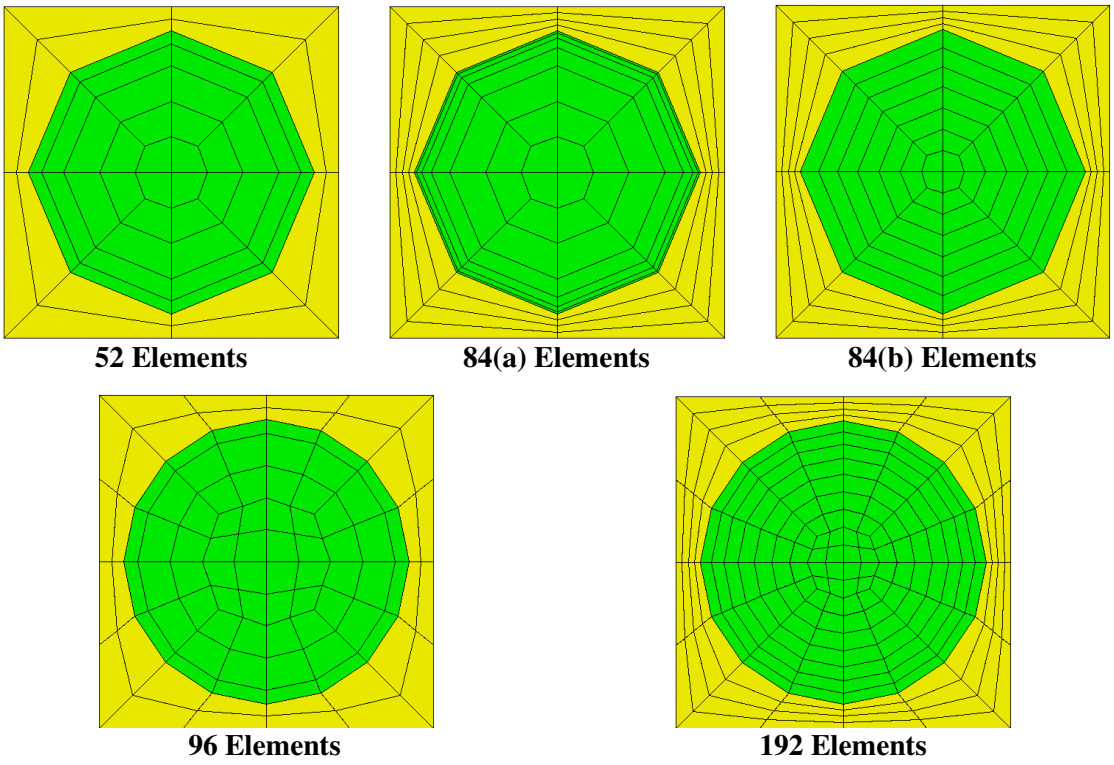
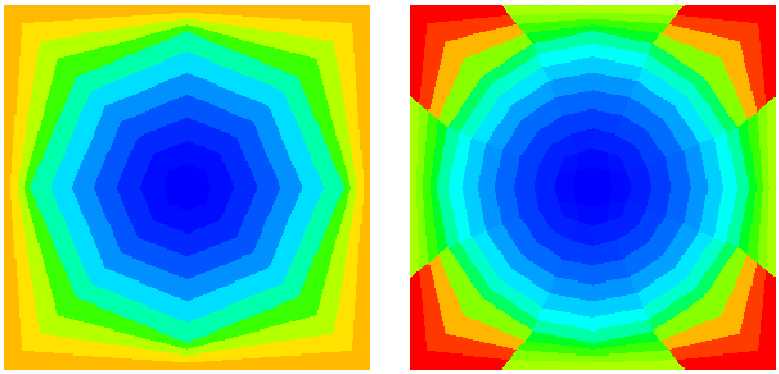


Figure IV. Finite Element Mesh Representations of a Single C5G7 Pin-cell.

Table 1. MOC Eigenvalue Results for a Single UO_2 Pin-cell

Elements	Eigenvalue
52	1.32624
84(a)	1.32620
84(b)	1.32621
96	1.32585
192	1.32574



84(b) Elements 192 Elements

Figure V. MOC Solution for the Seventh Energy Group Flux for a Single UO_2 Pin-cell.

Figure VI shows the C5G7 benchmark geometry where reflected boundary conditions are applied to the bottom and left and vacuum boundary conditions are applied to the top and right. As can be seen, half of the assemblies consist entirely of UO_2 fuel with the other assemblies containing MOX fuel. The steep global flux gradient combined with the heterogeneity of the individual assemblies makes this benchmark rather difficult to solve.

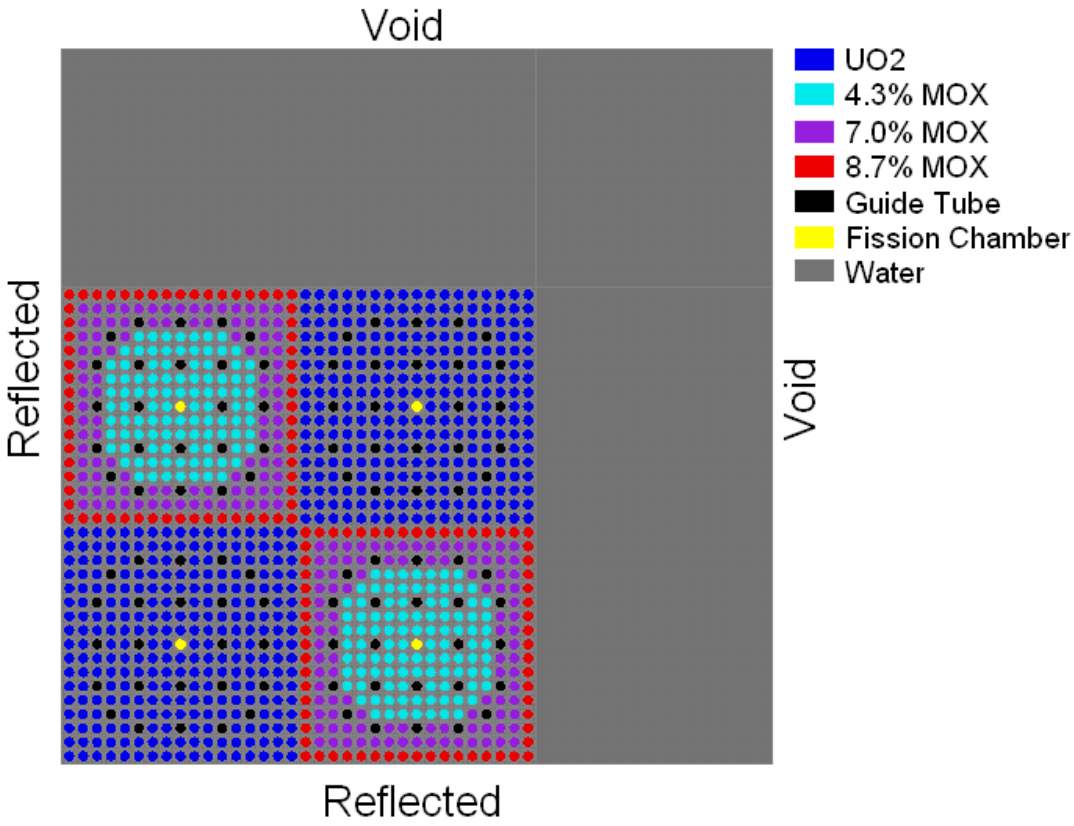


Figure VI. Two-Dimensional C5G7 Benchmark.

Table 2 gives the eigenvalue results for various combinations of mesh refinement, angular cubature, and trajectory spacing. The reference MCNP solution was given as 1.18655 ± 0.00010. The first set of data in

Table 2 considers refinements of the spatial mesh and, as was the case in the single pin-cell problem, a significant drop in the eigenvalue is observed with the change in the meshing scheme for the radial plane. As a consequence, we chose to use the mesh derived from the 96 element pin-cell mesh for the remaining study of the variation of the parameters. As can be seen, variation of the number of polar and azimuthal directions results in a relatively small improvement in the eigenvalue and refinements in the trajectory spacing has no discernable impact on the eigenvalue.

Table 2. Eigenvalue Results for the C5G7 Benchmark

Number of Pin-cell Elements	Total Number of Elements	Number of Polar Directions	Number of Azimuthal Directions	Maximum Trajectory Spacing (cm)	Eigenvalue
52 84(a) 84(b) 96 192	***65168*** ***102160*** ***102160*** ***120656*** ***231632***	3	32	0.01	1.18687 1.18681 1.18682 1.18641 1.18630
96	120656	***2*** ***3*** ***6***	32	0.01	1.18649 1.18641 1.18649
96	120656	3	***16*** ***32*** ***64***	0.01	1.18616 1.18641 1.18653
96	120656	3	32	***0.02*** ***0.01*** ***0.005***	1.18641 1.18641 1.18641

Figure VII shows example scalar flux plots for all seven groups of the C5G7 benchmark. In conclusion, these solutions are in good agreement with the reference MCNP solution with the remaining error being attributable to further refinements in the various parameters.

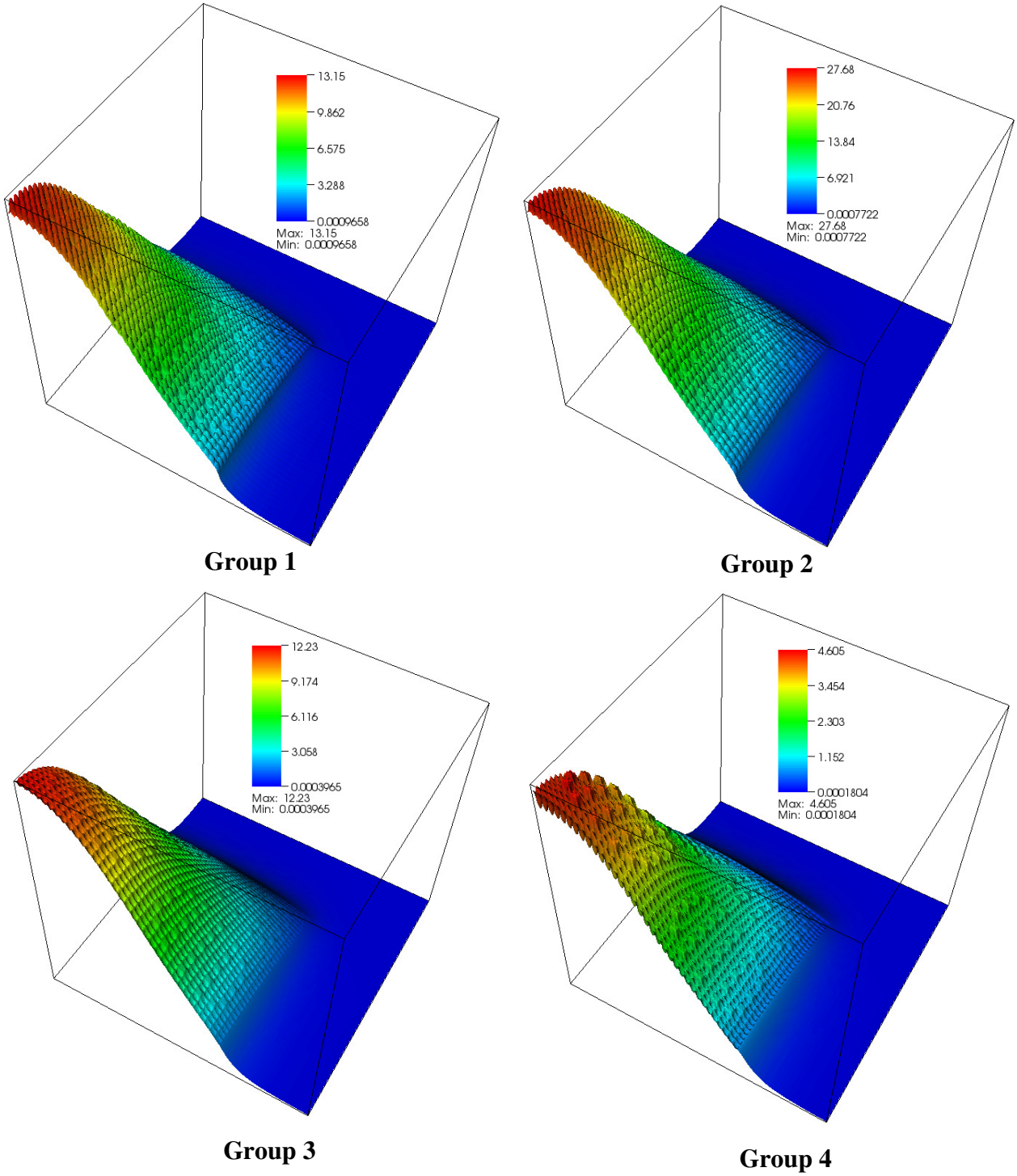
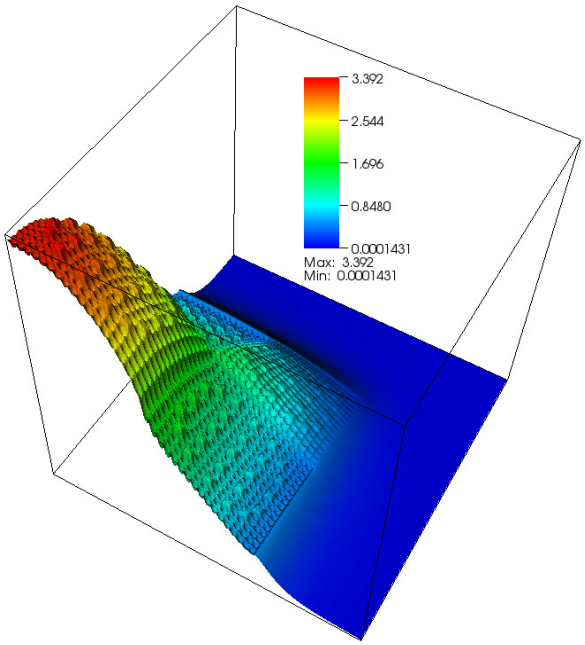
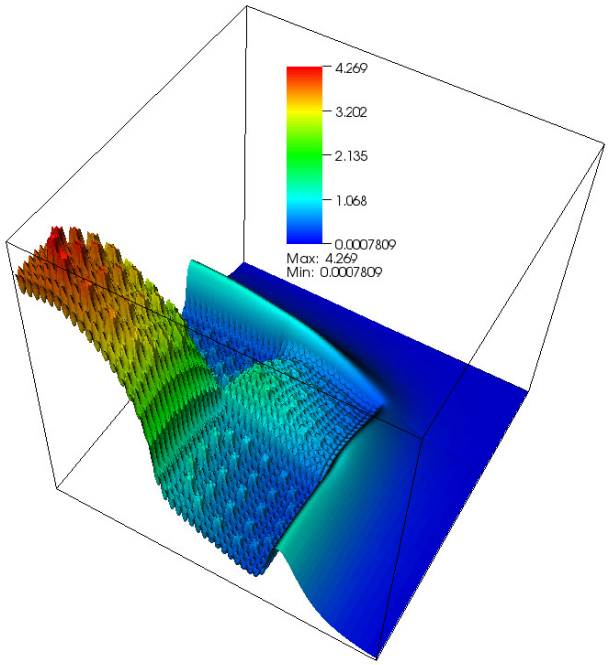


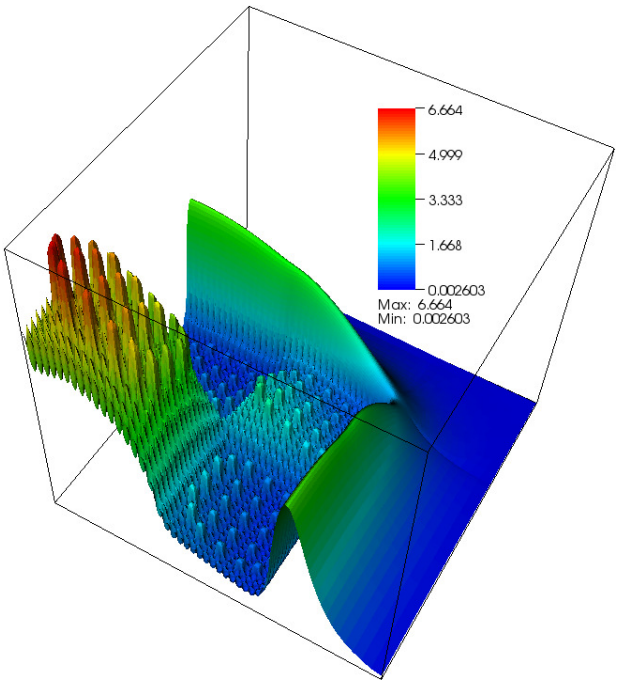
Figure VII. Example MOC Scalar Flux Solutions for the C5G7 Benchmark



Group 5



Group 6



Group 7

Figure VII. Example MOC Scalar Flux Solutions for the C5G7 Benchmark.

2.4 Parallel Performance of the MOCFE Solver

Part of the FY2008 work was to investigate more robust parallel algorithms for the MOCFE solver. The current parallel algorithm can be considered to consist of two initialization steps and four steps that are performed repeatedly during the iteration process.

Initialization

1. Given a list of trajectory starting points on the surface of the domain, find the intersections of all of the trajectories with all of the elements in the mesh for all angles in the cubature
2. Setup the coefficient matrix for the diffusion synthetic acceleration of the scattering iteration

Iterative Parallel Processes

1. For reflected boundaries of the mesh, perform a reduction followed by a broadcast over the global communicator on the boundary incident flux
2. Solve the propagation equation for the locally assigned trajectories to compute the contribution to the element averaged scalar flux and the contribution to the exiting boundary flux on reflected boundaries
3. Perform a reduction followed by a broadcast over the global communicator of the element averaged scalar flux
4. Solve the diffusion synthetic acceleration equation to update the scalar flux and perform a reduction followed by a broadcast over the global communicator of the correction.

The remaining parallel and sequential operations can be considered minor and/or very scalable in the current scheme and thus not important.

The current approach to distributing the trajectories (step 1 in Initialization) is done by first computing the total number of trajectories and approximately assigning an equal number of trajectories to each processor. As expected, a significant component of the computational effort is spent solving the propagation equation for each trajectory. The computational effort required for each trajectory is strongly dependent upon the number of intersections along each trajectory and thus the computational effort for each processor is strongly dependent upon the number of intersections that each processor finds along its share of trajectories. As such, the current approach to distributing the trajectories does not guarantee a perfect balance of the intersection information and we can observe a substantial computational load imbalance. One easy way to estimate the load imbalance is to consider the ratio of the maximum to minimum number of intersections in a given parallel job. Table 3 shows the computed ratios for the Takeda 4 benchmark [5] while Table 4 shows the computed ratios for a benchmark derived from the recent ABTR work [6].

Table 3. Computational Load Imbalance for the Takeda 4 Benchmark

Number of Processors	Angular directions	
	18	72
16	2.20	3.01
32	2.88	3.92
64	3.57	5.88

Table 4. Computational Load Imbalance for the ABTR Benchmark

Number of Processors	Angular directions	
	18	**72**
16	1.40	1.56
32	1.83	1.60
64	2.22	1.92

As can be seen, as the number of processors increases, the load becomes more unbalanced indicating that the current algorithm for dividing the computational work is insufficient. As it turns out, this problem has already been studied in the literature for MOC and is relatively easy to fix [7].

We have found that the steps 1, 3, and 4 of the iterative parallel processes are the primary bottlenecks of the MOCFE solver. From the preceding results of the two-dimensional OECD benchmark and several three-dimensional benchmarks including the Takeda 4 and ABTR benchmark we can trace the scalability limits of the current parallelization scheme to the flat source approximation currently implemented in each element. The flat source approximation inherently requires a large number of elements in order to guarantee solution accuracy which translates to large communication events in the global reduction and broadcast operations. These operations become progressively more expensive on most parallel machines as the vector becomes large and/or the number of processors increases. Consequently, the current parallel algorithm is impractical since we will invariably require large numbers of elements to solve the heterogeneous geometry problems that the MOCFE solver is targeting. Part of the remainder of FY2008 is focused on researching alternative parallel algorithms.

3. Development work for the PN2ND and SN2ND Solvers

As mentioned previously, the primary focus for the PN2ND and SN2ND solvers was to implement additional parallelization by angle and space. The intent was to create a grid communication pattern as outlined in Figure VIII. In this approach we assume the set of processors used in the calculation can be divided into three communicator sets corresponding to space (S), angle (A), and energy (G). For discussion purposes, we will use the notation (s,a,g) to represent a point in the grid shown in Figure VIII. The idea behind the communicators is to define a point to point communication process which can be performed simultaneously. Using an example, we can more easily explain the communication pattern and thus indicate the impacts on performance. First, we assume that the spatial set of vertices is partitioned into $\boldsymbol{S}$ pieces or segments, the angular directions (or moments for the PN2ND solver) are partitioned into $\boldsymbol{A}$ segments and the energy groups are partitioned into $\boldsymbol{G}$ segments. We use the word segment because we assume each space-angle-energy segment is contiguous in some manner with respect to the physical ordering of the degrees of freedom in a comparable sequential approach. If we assume that we need to communicate the space-angle information from energy segment 1 to energy segment 2, then we can write this as $(s,a,1)\rightarrow(s,a,2)$. If we have to perform this operation for all segments of s and a, then we can write the series of relations:

$$\begin{matrix} (1,1,1)\rightarrow(1,1,2) & (2,1,1)\rightarrow(2,1,2) & \cdots & (S,1,1)\rightarrow(S,1,2) \\ (1,2,1)\rightarrow(1,2,2) & (2,2,1)\rightarrow(2,2,2) & \cdots & (S,2,1)\rightarrow(S,2,2) \\ \vdots & \vdots & \ddots & \vdots \\ (1,A,1)\rightarrow(1,A,2) & (2,A,1)\rightarrow(2,A,2) & \cdots & (S,A,1)\rightarrow(S,A,2) \end{matrix},$$

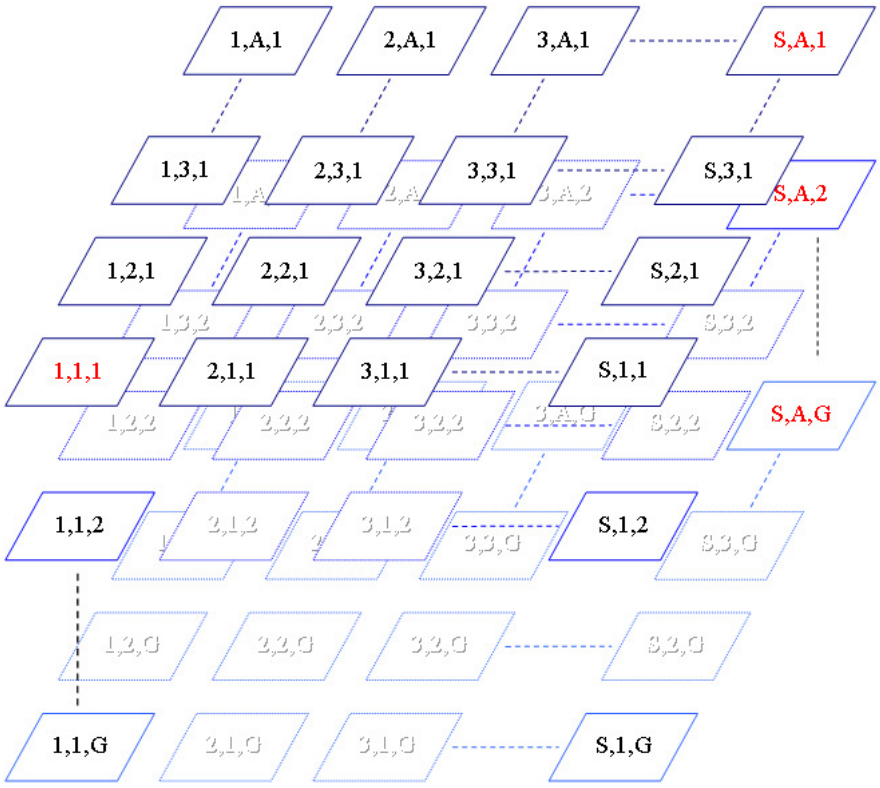


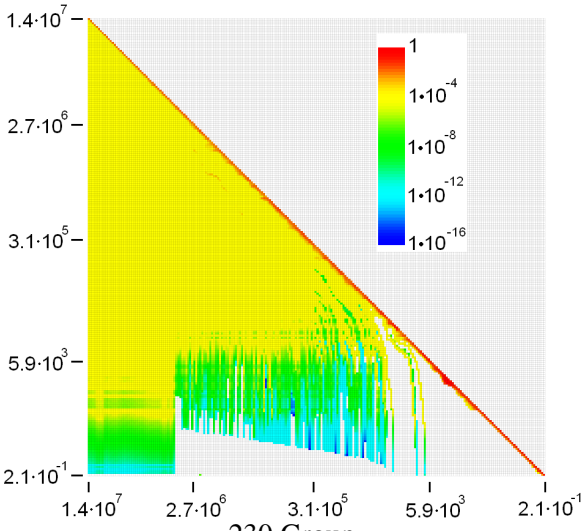
Figure VIII. Parallelization Strategy for the PN2ND and SN2ND Solvers

where each communication event can be performed simultaneously; a total of S•A communication events. To handle this in a parallel computing environment, we define S•A communicators where each processor rank is defined according to its order in the energy segmentation $\boldsymbol{g}$. A similar pattern exists when we want to transfer information in the other directions of the grid and we get an additional S•G+A•G communicators. Of course for each processor (s,a,g), only the three communicators relevant for the point to point communication are observable.

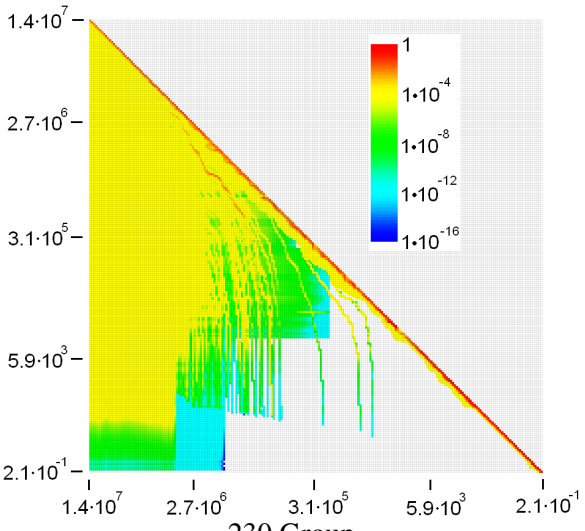
The discrete ordinates methods are typically written with a scattering iteration approach to the within group equation where a synthetic diffusion equation is used to accelerate the scattering iteration. This synthetic equation typically uses a single angular degree of freedom and we have consequently defined another communicator over the entire space-angle segmentation. The concept behind this is that we can spread the synthetic equation over the space-angle set of processors rather than just host it on one of the existing angular segments. This basically will force a further segmentation of the spatial approximation over each angular segment. The major drawback of this approach is that the algorithmic efficiency of the Krylov subspace solver tends to degrade when the spatial domain is broken up too finely. In our current implementation of the SN2ND coding we do not utilize the synthetic communicator and instead host the synthetic equation on the first processor on the angular communicator. The disadvantages with this approach are that the memory load on the first processor is doubled and the work load is unbalanced since the remaining processors on the angular communicator must wait until the synthetic equation is solved. The motivation behind the current approach was the reduced burden of coding required to get the solver setup.

As mentioned above, neither the PN2ND or SN2ND solvers use group segmentation at this point. The major reason for this is due to the scattering cross sections which couple the energy segments together.

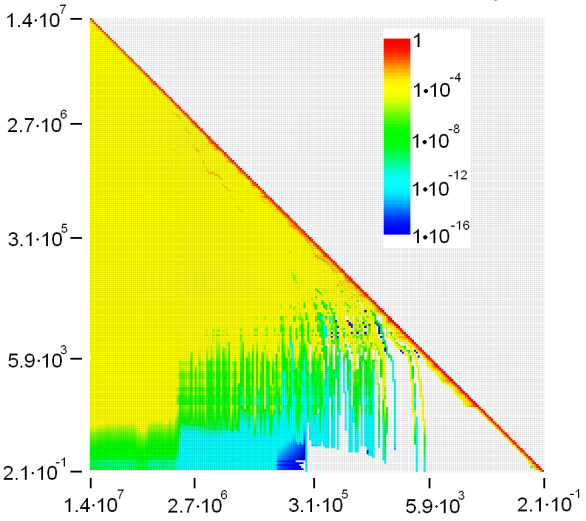
Figure IX provides example stenciling of the scattering cross section for some typical reactor problems we intend to use UNIC on. In each case a maximum normalization was applied separately for each figure and the energy group structure is quite different between the thermal reactor composition and the fast reactor compositions. As can be seen, the stenciling for the fast reactor compositions can vary substantially, however, it is still strictly lower triangular. The large upscattering region makes the thermal reactor problem dramatically different from the fast systems eliminating the strict lower triangular structure. The primary issue for devising an effective parallelization strategy here the lower triangular is nature of this system. The historical approach to solving this system is to start at the highest energy group and sweep down through each group which is equivalent to a back substitution algorithm. This approach is exact for the strict lower triangular approach, but for those problems with significant upscattering, an additional iterative scheme (upscatter iterations) must be employed to achieve the correct answer.



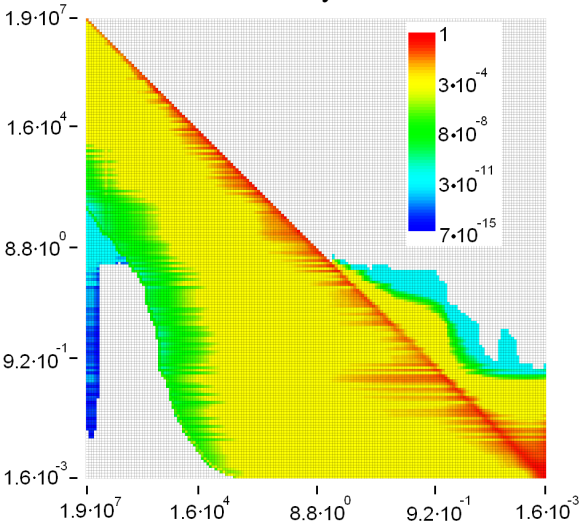
230 Group
Sodium Fast Reactor Fuel Assembly



230 Group
ZPPR15 Critical Assembly Structural Material



230 Group
Lead Fast Reactor Fuel Assembly



172 Group
PWR Fuel Assembly

Figure IX. Scattering Cross Section Stenciling

With regard to parallelization, the back substitution approach would require the processors assigned to the lower portion of the energy domain to wait until the solution on the highest energy portion of the energy

domain is obtained and communicated before continuing. Such an approach is obviously not a true parallel algorithm. From the literature, the comparable methods that exhibit this type of connection use some form of multigrid acceleration which is typically combined with a block Jacobi decomposition of the full domain. Additional research will be necessary to study similar algorithms for the energy domain if group segmentation is to be utilized. For now, all of the necessary communicators were put into place for the PN2ND and SN2ND solvers and, when possible, all subroutines were setup to operate in an environment with group segmentation in play.

3.1 Development Progress of the PN2ND Solver

The spherical harmonics approximation is one of the oldest approximations used to solve the transport equation [2]. The general approach is to expand the angular flux in terms of orthonormal spherical harmonic trial functions as done in Eqs. 3.1 and 3.2. Figure X graphically displays some selected angular trial functions from Eq. 3.1.

$$\psi_g^+(\vec{r},\hat{\Omega}) = \sum_{l=0,2,\ldots,N-1} \sum_{m=-l}^{l} Y_{l,m}(\hat{\Omega})\psi_{g,l,m}^+(\vec{r}) = Y_+^T(\hat{\Omega})\psi_g^+(\vec{r}), \tag{3.1}$$

$$\psi_g^-(\vec{r},\hat{\Omega}) = \sum_{l=1,3,\ldots,N} \sum_{m=-l}^{l} Y_{l,m}(\hat{\Omega})\psi_{g,l,m}^-(\vec{r}) = Y_-^T(\hat{\Omega})\psi_g^-(\vec{r}), \tag{3.2}$$

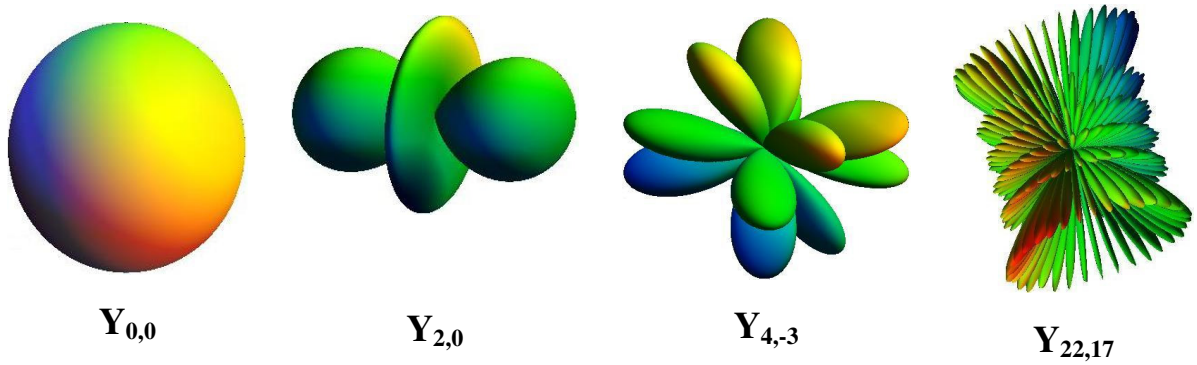
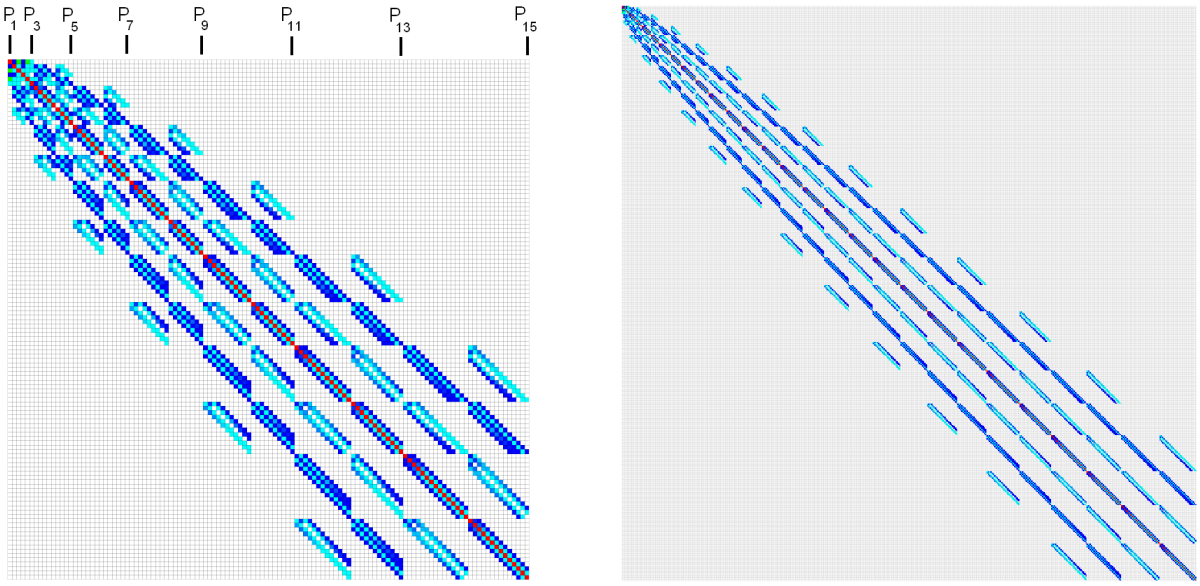


Figure X. Example Angular Trial Functions Utilized in PN2ND

While these trial functions are not coupled for the scattering, collision, or fission source terms (orthonormal) in the transport equation, they are coupled via the streaming operator. Figure XI shows the connection of the angular terms for a standard even-parity spherical harmonics formulation. As can be seen, the angular connections are symmetric and relatively sparse with approximately the same amount of non-zero data focused on the coupling within a given *L* order of spherical harmonic data and between two consecutive *L* orders of spherical harmonic data. It is important to note that there is no coupling between spherical harmonic terms that are more than two *L* orders apart (P_{15} is not directly connected to P_{11}) in the streaming operator. This coupling behavior is generally advantageous in a parallel algorithm since the connected portion of the angular flux on any given processor is relatively small compared with the total angular flux. However, this requires the assumption that a relatively high order angular approximation is going to be utilized, P_{11} or higher, which was never the targeted development goal of the PN2ND solver. Furthermore, when vacuum boundary conditions are applied to a boundary surface, the angular stencils shown in Figure XI become fully coupled. This greatly complicates any type of Krylov solver operations that are to be performed since it requires that all of the angular moment data be present on any given processor to correctly compute the action of the coefficient matrix. The reflected boundary conditions are not as much of a problem as the vacuum boundary conditions, but they do increase the non-zero filling of the existing stencil shown in Figure XI.



P_{15} **P_{29}**

Figure XI. Example Three-Dimensional Angular Stencils for PN2ND

Figure XII gives a typical spatial stencil that is observable for a single processor in a parallel computation. The mesh consists of ~1400 spatial vertices and 300 quadratic finite elements. As can be seen, the stencil does not exhibit the typical banded matrix structure that one would expect, primarily because we do not currently apply a bandwidth optimization algorithm in PN2ND or SN2ND. Given that we are already using an efficient sparse matrix vector multiplication routine, we do not intend to study the optimized bandwidth approaches until a full algorithmic performance can be made of the existing solvers.

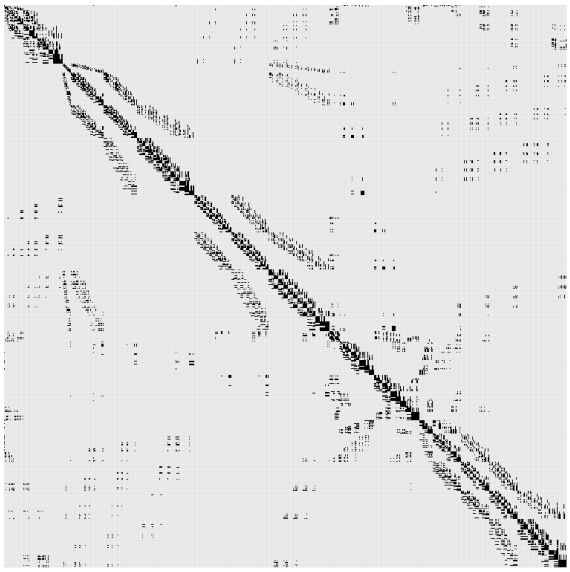


Figure XII. Example Spatial Stencil (Not Bandwidth Optimized)

As mentioned previously, the current UNIC code focuses on solving the within group transport equation which inherently requires that the space-angle system be solved simultaneously. To handle this, the space and angular variables in the PN2ND solver are tensored together to form a single vector. To visualize this,

we included an example P_5 coefficient matrix stenciling in Figure XIII. Note that for each non-zero angular moment coupling derived from Figure XI, the spatial matrix stencil of Figure XII is present.

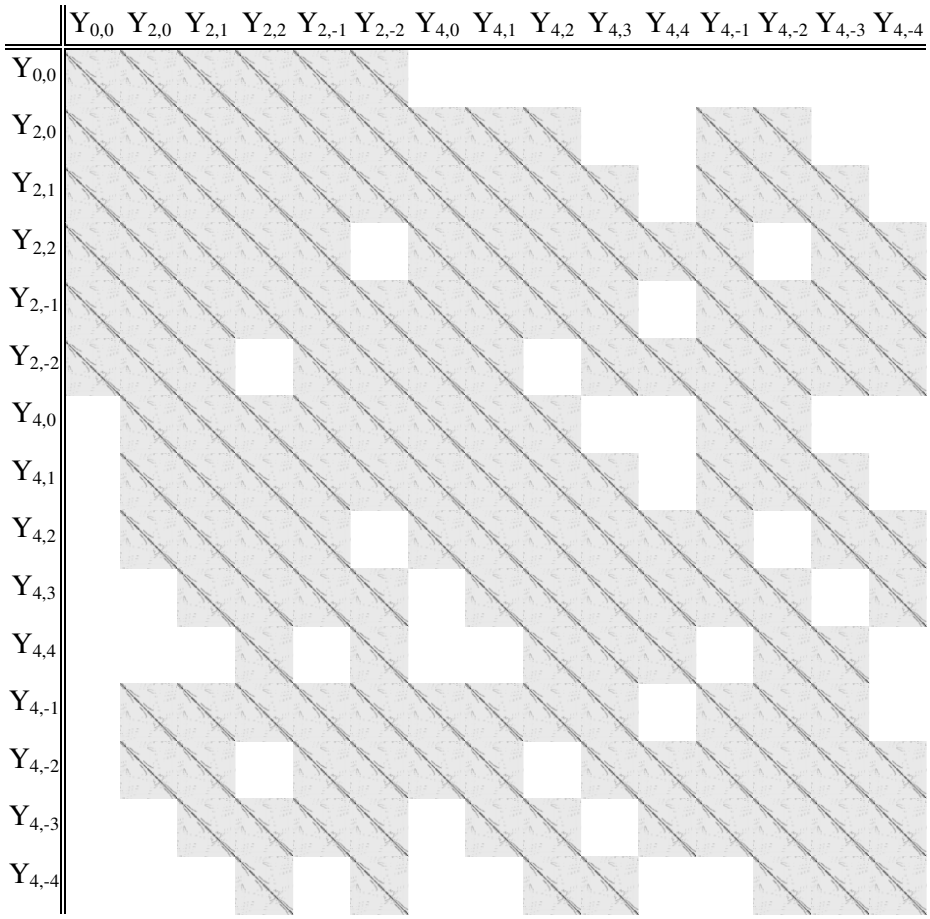


Figure XIII. P_5 Space-Angle Stencil

In the previous version of the PN2ND solver, we found that the large memory requirements derived from the storage outlined in Figure XIII prevented the method from being useful on most of the parallel machines. As a consequence of this, we choose to change the preconditioner in the PN2ND solver to block Jacobi where either incomplete Cholesky or SSOR is used in each angular block. For Figure XIII, this preconditioner would define 15 angular blocks that can be solved simultaneously. As one would expect, block Jacobi is a much poorer preconditioner than the previous one, SSOR or incomplete Cholesky on the entire space-angle system, and thus the number of iterations will increase for the new preconditioner.

Our goal of course was to be able to overcome the reduction in the effectiveness of the preconditioner by relying upon the fact that more processors would be available in the parallel environment. As it turns out, homogenous problems typically require relatively few spatial elements and thus few spatial vertices. Both previous versions of PN2ND and SN2ND solvers could not be scaled to more than 20-50 processors on small benchmarks such as the Takeda [1], ZPPR15, and the 30 degree ABTR benchmark [1]. In general, the segmenting of the spatial domain is limited by the ratio of the number of spatial vertices lying on the surface of the local processor's mesh to the total number of spatial vertices in the local processor's mesh (i.e. surface area to volume ratio). For three-dimensional meshes with quadratic elements, using less than 1000 spatial vertices leads to a substantial degradation in the parallel performance of the Krylov solver. While the new preconditioner does require more iterations to be performed, the ability to scale by angle

and space can yield a net improvement in the performance given even modest scalability with respect to angle. As an example, the Takeda 1 benchmark [5] only requires ~30000 spatial degrees of freedom and 120 angular degrees of freedom (P_{15}). The old PN2ND solver can only scale to ~30 processors (1000 per processor) and, because of memory limitations, can only apply a P_7 approximation in angle. Contrary to this, the new version can solve beyond P_{15} on the same 30 processors, and, further, it can theoretically scale to 3600 processors given its ability to solve the angular blocks simultaneously. Note that all of this has yet to actually be tested out.

In addition to the above work on improving the parallel scalability of the PN2ND solver, the introduction of a block Jacobi preconditioner required a modification of the reflected boundary condition treatment in the PN2ND solver. While the previous version did work, it was prone to problems because of the ad hoc scaling factors that were introduced as discussed in Appendix A. In the new solver, the ambiguities in the solution scheme are removed and thus the reliability of the PN2ND solver has improved.

3.2 Benchmark Problems Solved Using the New PN2ND Solver

While the new version of the PN2ND solver was completed in February, the iterative algorithm and block Jacobi preconditioner have not yet been optimized. Since we still have to implement the Tchebychev acceleration and we need to optimize the various components (the remaining work to be performed in FY2008), we have not performed a wide range of benchmark problems with the new solver. The only new benchmark problem that was attempted using the new PN2ND solver is a homogenized drawer representation of the ZPPR15 critical assembly shown in Figure XIV.

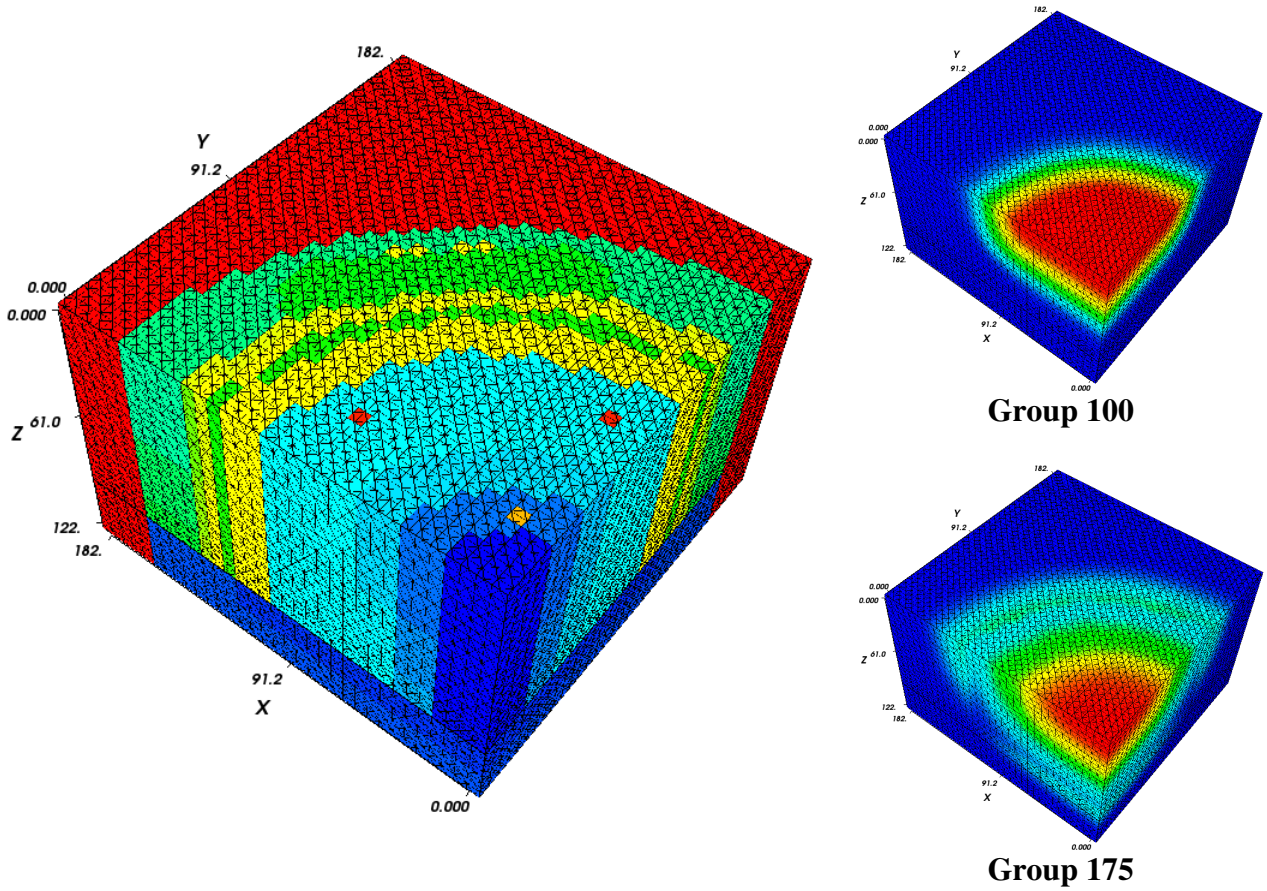


Figure XIV Geometrical Layout and Example Flux Solutions of the ZPPR15 Benchmark.

The primary difficulty of this benchmark is that it utilizes a P_3 scattering kernel and a 230 group cross section library. Both the new and old PN2ND solvers were used on the new COSMEA cluster which has 128 processors and 4 GB of memory per processor. As it turns out, the old PN2ND solver could only be applied for P_1 and P_3 angular approximations with a P_5 approximation requiring too much memory. Further, the old PN2ND solver could not be combined with an anisotropic scattering kernel higher than P_1 due to a yet unknown bug and thus the results are omitted here. Contrary to this, the new PN2ND solver had no significant memory issues and appears to be able to go up to at least P_9 on COSMEA. However, given the lack of an optimized iterative algorithm, the computational time is high and only P_1 through P_5 angular approximations were attempted where a P_3 anisotropic scattering kernel was utilized in the P_3 and P_5 calculations**.** Table 5 gives the PN2ND solver eigenvalue solutions for a mesh with 76,000 spatial vertices along with the VIM solution of the plate-by-plate geometry model. We note that a substantial amount of the "accuracy" of the PN2ND solver relies upon the slab geometry cell calculations performed in SDX to generate the homogeneous coefficients. We also attempted to get comparative solutions to this benchmark using the VARIANT nodal transport option in DIF3D [8,9], but some memory issues with anisotropic scattering could not be overcome in that code.

Table 5. PN2ND Solver Results for the ZPPR15

Input Settings	Eigenvalue
P_1-P_1	0.99258
P_3-P_3	0.99640
P_5-P_3	0.99651
Monte Carlo (VIM)	0.99616±0.00010

It is obvious from this benchmark is that a P_5 angular approximation is sufficient to converge angularly which is not typical of most of the other benchmark problems we have performed with this solver. This is primarily due to the size of the ZPPR15 and the presence of depleted uranium blankets both of which reduce the importance of leakage out of the system.

3.3 Development Progress of the SN2ND Solver

While the preceding PN2ND solver developments allows us to apply it to a wider range of problems, the primary focus of introducing the parallelization by angle was the discrete ordinates methods. The motivation for this approach requires a close inspection of the system of equations. Eq. (3.3) gives the compact form of the within group flux equation for the SN2ND solver.

$$\left(A \ -N\right)\psi^{+} = S^{+} \tag{3.3}$$

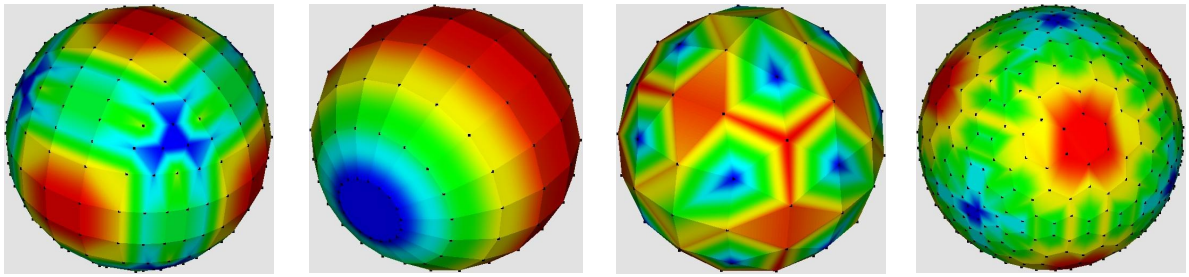
In Eq. (3.3), ψ^{+} represents the even parity discrete ordinates flux, A is a sparse symmetric, positive definite coefficient matrix resulting from discretization of the streaming operator, the collision operator, and the boundary conditions, while N is an unsymmetric, dense coefficient matrix resulting from the within group scattering operator. We typically choose to solve this set of equations using a scattering iteration which introduces the iterative indices i as shown in Eq. (3.4).

$$A \ \ \psi^{+}_{i+1} = S^{+} + N\,\psi^{+}_{i} \tag{3.4}$$

To fully understand the relation between the discrete ordinates flux moments and the spherical harmonic based scattering kernel, we need a few auxiliary relationships. First, we define a set of directions on the surface of the sphere, $\hat{\Omega}_n$, and assign weights, w_n, to these directions such that we can write

$$\varphi^{+}_{l,m}(\vec{r}) = \int Y_{l,m}(\hat{\Omega})\psi^{+}(\vec{r},\hat{\Omega})d\Omega = \sum_{n} Y_{l,m}(\hat{\Omega}_n)\psi^{+}(\vec{r},\hat{\Omega}_n)w_n \ , \tag{3.5}$$

where $\varphi^{+}_{l,m}(\vec{r})$ represents the even-parity spherical harmonics projection of the angular flux to the spherical harmonic function $Y_{l,m}(\hat{\Omega})$. Figure VIII shows some example angular cubatures that satisfy Eq. (3.5), where the directions are defined as the set of black points on the surface of the sphere and the coloring indicates the variation of the weight assignment.



Carlson Level-Symmetric S_{16} | **Square Legendre-Tchebychev S_{16}** | **Lebedev-Laikov S_{15}** | **Thurgood S_{15}**

Figure XV. Example Angular Cubature for the Discrete Ordinates Method

Using this approach we find that the coefficient matrix *A* is generally block diagonal as outlined in Figure XVI for a level-symmetric S_2 angular cubature in three-dimensional geometries. This block diagonal nature is only altered when reflected boundary conditions are applied. To handle reflected boundary conditions the set of the angular directions is partitioned into "dependent" and "independent" directions, where the set of dependent directions can be written as functions of the independent directions. The imposition of these constraints in Figure XVI eliminates the strict block diagonal nature for all angular directions of those vertices that lie upon the reflected boundary condition.

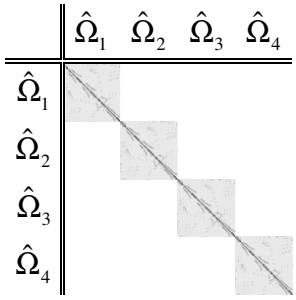


Figure XVI. P_5 Space-Angle Stencil

As discussed previously, we have chosen not to use a space-angle connectivity assignment because of the problems it causes in the parallel framework. Instead, we choose to augment Eq. (3.4) into the set of equations given by Eq. (3.6), where the matrix T_{ref} is described in Appendix B.

$$\begin{aligned}
&\tilde{\psi}^{+}_{i+\frac{1}{2}} = \mathrm{T}_{ref}\psi^{+}_{i} \\
&\mathrm{T}_{ref} A\ \mathrm{T}^{T}_{ref}\ \tilde{\psi}^{+}_{i+\frac{1}{2}} = \mathrm{T}_{ref} S^{+} + \mathrm{T}_{ref} N\psi^{+}_{i} \quad \rightarrow \quad \tilde{A}\,\tilde{\psi}^{+}_{i+\frac{1}{2}} = \tilde{S}^{+}_{i} \\
&\psi^{+}_{i+1} = \mathrm{T}^{T}_{ref}\tilde{\psi}^{+}_{i+\frac{1}{2}}
\end{aligned} \tag{3.6}$$

The first relation in Eq. (3.6) constricts the full angular space to the space only containing independent directions by utilizing the relations between the independent and dependent set of directions defined as T_{ref}. The second relation compacts the space of the coefficient matrix which we feed into the parallel Krylov solver. The third relation is used to define the solution for the dependent directions using the relations that define them in terms of the independent directions. Given the extremely sparse nature of the T_{ref} matrix, and the ease of computing the coefficient matrix *A*, this approach is more numerically more robust when compared with the previous approach used in the SN2ND solver. We also note that this method does not require the ad hoc factors that appear in the PN2ND solver given that Eq. (3.6) imposes a direct partitioning of the independent and dependent angular degrees of freedom. The only complicated part of this method is the communication costs of updating the within group scattering contribution and properly treating the reflected boundary conditions.

To begin, we assume a problem where we have fully segmented the angular space such that each processor is responsible for a single angular direction. To construct a volumetric source for a given direction $S^{+}(\vec{r},\hat{\Omega}_n)$ we take the source derived from the even parity flux $S^{+}_{l,m}(\vec{r})$ and project it to the discrete ordinates space using

$$S^{+}(\vec{r},\hat{\Omega}_n)=\sum_{l,m} Y_{l,m}(\hat{\Omega}_n)S^{+}_{l,m}(\vec{r})\,. \tag{3.7}$$

With this we can expand the within group scattering operation *N* in Eq. (3.6) as

$$N=\left(N_{+}\otimes I_s\right)\left(\sum_{K,L}\left\{V_K^T\tilde{\sigma}_{-}V_L\otimes U_K F^{-1}U_L\right\}+\tilde{\sigma}_{+}\otimes F\right)\left(M_{+}\otimes I_s\right). \tag{3.8}$$
$$N=N_{+}A_s M_{+}$$

where $\tilde{\sigma}_{\pm}$ are derived from the within group scattering cross sections, M_{+} is equivalent to the operation in Eq. (3.5), and N_{+} is equivalent to the operation defined in Eq. (3.7). The matrix A_s represents the within group scattering operation which has a space-angle stencil identical to that seen in the $\mathrm{P_N}$ system in Figure XIII. The subtle part of Eq. (3.8) is that it is fundamentally a series of non-square matrix because the number of spherical harmonics used in the scattering kernel is typically far less than the number of angular directions used in the cubature. Consequently, communicating the spherical harmonics based flux versus the discrete ordinates flux can translate to a huge difference in the size of the data that is to be communicated.

The most efficient approach with regard to communication is to assume each processor must be provided the source in the spherical harmonic space. Further, we assume that the scattering operations that must be performed in A_s can be distributed on the existing angular communicator. We then ask each processor to perform its operations on the local column space of M_{+} to obtain the contribution to all of the spherical harmonic moments

$$\varphi^{+}_{l,m,n}=M^{n}_{+}\psi^{+}_{n}. \tag{3.9}$$

Then, we communicate the flux moments on the angular communicator such that the locally owned moments of the spherical harmonic flux are obtained (scatter-gather). We can then perform the scattering kernel operations in A_s for only the locally owned column space of A_s (i.e. the locally owned spherical harmonic moments) to obtain

$$s^{+}_{l,m} = A^{l,m}_{s}\ \varphi^{+}_{l,m}\,. \tag{3.10}$$

Now we perform a global reduction operation on the subset of processors such that all processors on the angle communicator receive the fully updated set of moments s^{+}. With this information, we can locally construct the directional source for the locally owned angular moment of the flux by applying the local column space of N_{+}.

$$N\psi^{+} = N^{n}_{+}\ s^{+}\,. \tag{3.11}$$

This approach poses the minimal requirements for communication and the maximal possible parallelization of the scattering kernel operations given that we are segmenting the angular space. The reflected boundary conditions complicate this procedure somewhat; however, given that we have fundamentally transmitted the moments of the source given by Eq. (3.10) to all processors in the angle communicator, we only have issues during the Krylov solver operation involved to update $\tilde{\psi}^{+}_{i+\frac{1}{2}}$ in Eq. (3.6). The communication costs of this operation are associated with the non-zero connection in the T_{ref} matrix. For three-dimensional problems and angular cubatures that meet the symmetry conditions of the domain it is easy to show that we will see a maximum of four and an average of one non-zeros per row. When the cubature doesn't meet the symmetry of the domain we have observed up to 10-20 non-zeros per row when the angular cubature contains hundreds of directions. To properly solve for $\tilde{\psi}^{+}_{i+\frac{1}{2}}$ we have to communicate the angular flux moments that are connected, which means that we must communicate a subset of the angular flux for those vertices that lie on reflected surfaces. At worst we could simply communicate the angular flux for those vertices that lie on the reflected surface. Because this constrains the number of spatial degrees of freedom we can simply employ a broadcast operation of all moments of the angular flux for these vertices on the angular communicator.

The implementation of this procedure is obviously quite complicated, and for the short term we have chosen to simply broadcast the angular flux for all spatial vertices on the angle communicator and duplicate the scattering source operations. In future work we intend to fully implement the preceding minimal communication approach as time permits. It is important to note that even though our approach is more expensive than the optimal one, we are still imposing less communication than that utilized in the previous version of the SN2ND solver and we should therefore see some improvement.

Another issue is how to distribute the synthetic equation used to accelerate the scattering iteration defined by Eq. (3.6). As mentioned earlier, we have chosen to host this equation on the first processor of the angular communicator which causes a load imbalance during the solve process. Before we take the approach where we distribute this equation over the entire angle communicator, we want to investigate how much parallelism we have achieved with the existing approach and the minimal communication approach. In short, we believe the solution of the synthetic equation will prove to be a type of multigrid acceleration scheme and, to avoid the degradation of the Krylov solver, we will likely always prefer to host the synthetic equation on the first processor or some subset of the processors in the angular communicator.

At present, the new SN2ND solver with parallelism by angle has been implemented in UNIC and is undergoing final debugging and validation. When the new version is validated, we will implement the Tchebychev acceleration scheme and begin the optimization and benchmarking study that is scheduled to be performed for the remainder of FY2008. We will also consider implementing the minimal communication approach outlined above which will is independent of the Tchebychev and iterative optimization study.

4. Conclusions and Future Work

A two-dimensional geometry capability was added to the MOCFE solver. As seen, the solver works sufficiently well to produce very good solutions to a complicated and difficult benchmark problem. As expected, the current parallel algorithm used in the MOCFE solver is not scalable for either the two- or three-dimensional geometry options of the MOCFE solver. Our goal for this year is to do further research on the MOC method to develop a scalable algorithm.

The PN2ND and SN2ND solvers were updated to treat parallelization by angle and energy. Although the ability of either solver has not been thoroughly tested out at this point, it is clear that the memory burden in the PN2ND solver was greatly reduced. The goal for the remainder of the fiscal year is to implement Tchebychev acceleration in both solvers and optimize the iterative routine. A series of reactor type benchmark problems will be created to test out the new solvers and validate the capabilities on the existing parallel computing resources. In parallel with this work, we will be developing a kinetics capability to work with the UNIC solver. This process will require the development of an efficient parallel cross section storage format and the inclusion of an efficient fixed source iteration algorithm for the solvers in UNIC. Our goal is to setup the kinetics capability and solve some simple benchmark problems by the end of the fiscal year.

We also hope to create another solver in UNIC termed SN1ST. This solver is based upon the first order discrete ordinates method and is similar to SN2ND. However, unlike standard sweeping discrete ordinates methods for which scalability is still an open area of research, this solver will rely upon the proven capabilities of the GMRES solver in PETSc. As was the case with PN2ND and SN2ND, SN1ST will be able to scale to thousands of processors, but it will utilize less memory than either the PN2ND and SN2ND solvers. The primary motivation for creating this solver is the inability of the PN2ND and SN2ND solvers to handle problems with voids or pure scatter regions in them.

References

1. Micheal A Smith, Cristian Rabiti, Dinesh Kaushik, Won Sik Yang and Giuseppe Palmiotti, “Report on Advanced Neutronics Code Development FY2007,” ANL-AFCI-209, Sept. 2007.
2. E. E. Lewis and Jr. W. F. Miller. Computational Methods of Neutron Transport. Wiley, 1984.
3. C. Rabiti, M. A. Smith, G. Palmiotti, “A Three-dimensional Method of Characteristics on Unstructured Tetrahedral Meshes,” *Trans. Am. Nucl. Soc*. **96**, 470 (2007).
4. M. A. Smith, G. Palmiotti, et al., “Benchmark on Deterministic Transport Calculations Without Spatial Homogenization (MOX Fuel Assembly 3-D Extension Case),” OECD/NEA document, NEA/NSC/DOC(2005)16, October, 2005.
5. T. Takeda and H. Ikeda, “3-D Neutron Transport Benchmarks,” NEACRP-1-300 OECD/NEA, Organization of Economic Cooperation and Development/Nuclear Energy Agency (March 1991).
6. Y. I. Chang, P. J. Finck, and C. Grandy. Advanced burner test reactor preconceptual design report. Technical Report ANL-ABR-1 (ANL-AFCI-173), Argonne National Laboratory, Argonne, IL, 2006.
7. Mohamed Dahmani and Robert Roy, “Scalability Modeling for Deterministic Particle Transport Solvers,” International Journal of High Performance Computing Applications **20**, p 541 (2006).
8. R. D. Lawrence, “Progress in Nodal Methods for the Solution of the Neutron Diffusion and Transport Equations,” *Prog. Nucl. Energy*, **17**, 271 (1986).
9. G. Palmiotti, E. E. Lewis, C. B. Carrico, “VARIANT: VARIational Anisotropic Nodal Transport for Multidimensional Cartesian and Hexagonal Geometry Calculation,” Argonne National Laboratory ANL-95/40, 1995.

Appendix A

PN2ND Reflected Boundary Condition Treatment

A-1. PN2ND Reflected Boundary Condition Treatment

To correctly impose the reflected boundary conditions on a second-order spherical harmonics discretization of the transport equation, we must define a set of functions in the space of $\hat{\Omega}$ which satisfies the symmetry requirements of the reflected boundary conditions. When we combine this approach with the finite element discretization of the spatial domain we find that we are required to impose these constraints at every spatial vertex that lies on the surface of the reflected boundary condition(s). For those vertices that lie on apexes of multiple reflected boundary conditions we must impose all of the symmetry conditions which can at worst reduce the system of angular trial functions to an isotropic flux representation. To demonstrate the approach used in PN2ND, we use a simplified derivation starting with the system of equations

$$A\ \psi^{+} = S^{+}, \tag{A.1}$$

where ψ^{+} is the even-parity flux, A is the spherical harmonic coefficient matrix for the within group even-parity flux and S^{+} is the fixed source, fission source, and in scatter contribution for the current group. In this system of equations we assume that all vacuum (or void) boundary conditions are accounted for and all that is left is the imposition of the reflected boundary conditions. For a reflected boundary surface, the reflected boundary condition imposes symmetry conditions on the set of angular trial functions and we can view this as a truncation of the existing set of trial functions that we represent as:

$$Y_{+,ref}\left(\hat{\Omega}\right)\psi^{+}_{ref} = \tau_{ref} Y_{+}\left(\hat{\Omega}\right)\psi^{+}, \tag{A.2}$$

where τ_{ref} is a non-square matrix with respect to the number of angular trial functions in the vector $Y_{+}\left(\hat{\Omega}\right)$. As an example, Eq. (A.3) shows the matrix for a P_3 implementation three-dimensional boundary surface with a (1,0,0) outward normal.

$$\tau_{(1,0,0)} = \begin{matrix} Y_{0,0} & Y_{2,0} & Y_{2,1} & Y_{2,2} & Y_{2,-1} & Y_{2,-2} \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.3}$$

We can implement Eq. (A.2) into Eq. (A.1) to obtain

$$\begin{aligned} \mathrm{T}_{ref} A\, \mathrm{T}^{T}_{ref} \mathrm{T}_{ref}\, \psi^{+} &= \mathrm{T}_{ref} S^{+} \\ A_{ref}\, \psi^{+}_{ref} &= S^{+}_{ref} \end{aligned} \tag{A.4}$$

where T_{ref} is a non square Boolean type matrix. For those vertices that are not along the reflected boundary condition, the corresponding angular component of T_{ref} is an identity matrix. For those vertices that are on the reflected boundary condition, the angular truncation matrix in Eq. (A.3) is used. We note that not only is the angular flux truncated to satisfy Eq. (A.2), but the set of trial functions that

were used to weight the system of equations is also truncated, thus the presence of T_{ref} on both sides of the coefficient matrix in Eq. (A.4).

The primary problem with this approach is that we must define a space-angle connectivity list to handle the reduced set of trial functions that occurs. This causes problems with the parallelism of the solver and is fundamentally undesirable given that the source operation is substantially complicated. As a consequence, we formulated an alternative approach which obviates the need to truncate the system. To start we first define τ_{ref} in Eq. (A.3) to be square as shown in Eq. (A.5).

$$\tau_{(1,0,0)} = \begin{bmatrix} Y_{0,0} & Y_{2,0} & Y_{2,1} & Y_{2,2} & Y_{2,-1} & Y_{2,-2} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.5}$$

We then define the angular matrix π_{ref} as

$$I - \tau_{ref}^{T}\tau_{ref} = \pi_{ref} \,. \tag{A.6}$$

We can add this onto both sides of a modified Eq. (A.4) to get

$$\begin{aligned} \left(\mathrm{T}_{ref} A\, \mathrm{T}_{ref}^{T} + f\cdot\Pi_{ref}\right)\psi_{ref}^{+} &= \mathrm{T}_{ref} S^{+} + \left(f\cdot\Pi_{ref}\right)\psi_{ref}^{+} \\ A_{ref}\,\psi_{ref}^{+} &= \mathrm{T}_{ref} S^{+} + \left(f\cdot\Pi_{ref}\right)\psi_{ref}^{+} \end{aligned}, \tag{A.7}$$

The matrix Π_{ref} only contains π_{ref} for those vertices that lie on the reflected boundary conditions. The number f is an arbitrary scaling factor that has two purposes:

- Prevent the coefficient matrix in Eq. (A.7) from being singular
- Produce the best possible condition number

As can be seen, this introduces an iterative procedure similar to a scattering iteration in a discrete ordinates method. The problem with the old version of PN2ND was that this iterative procedure was not incorporated because the definition of Π_{ref} was not explicitly maintained and thus the solution could be perturbed by the magnitude of $\left(f\cdot\Pi_{ref}\right)\psi_{ref}^{+}$. This error was typically << 20 pcm for most benchmark problems, and only appeared for problems with non-Cartesian boundary conditions like the 1/6 symmetry ABTR benchmark and the Takeda 4 benchmark. In the new version, we implemented a matrix free approach to the A matrix such that we can implement the iterative solution of Eq. (A.7).

A-2. Reflected Boundary Condition Performance Issues

In general, we have found that this equation requires at most two iterations for a wide range of problems given a reasonable selection of f. The magnitude of the scaling factor should be linked to the magnitude of the diagonal term of A for the targeted spatial vertex. However, variation of the cross sections for a smaller test problem indicated nearly consistent computational performance with a fixed scaling factor of 10^{-3}. This is likely due to the fact that the additive term is decoupled from the remainder of the system and we are in effect defining the relative magnitude of this additive term to the existing system for which 10^{-3} is not an unreasonable constant.

We also performed an additional study on the impact of the scaling factor on computational time for the Takeda 4 benchmark [5]. In the new PN2ND solver, we can separate reflected boundary conditions that are pure Cartesian boundary conditions from those that are non-Cartesian and thus apply different scaling factors to both. Figure XVII shows the computational performance of PN2ND solving the Takeda 4 benchmark.

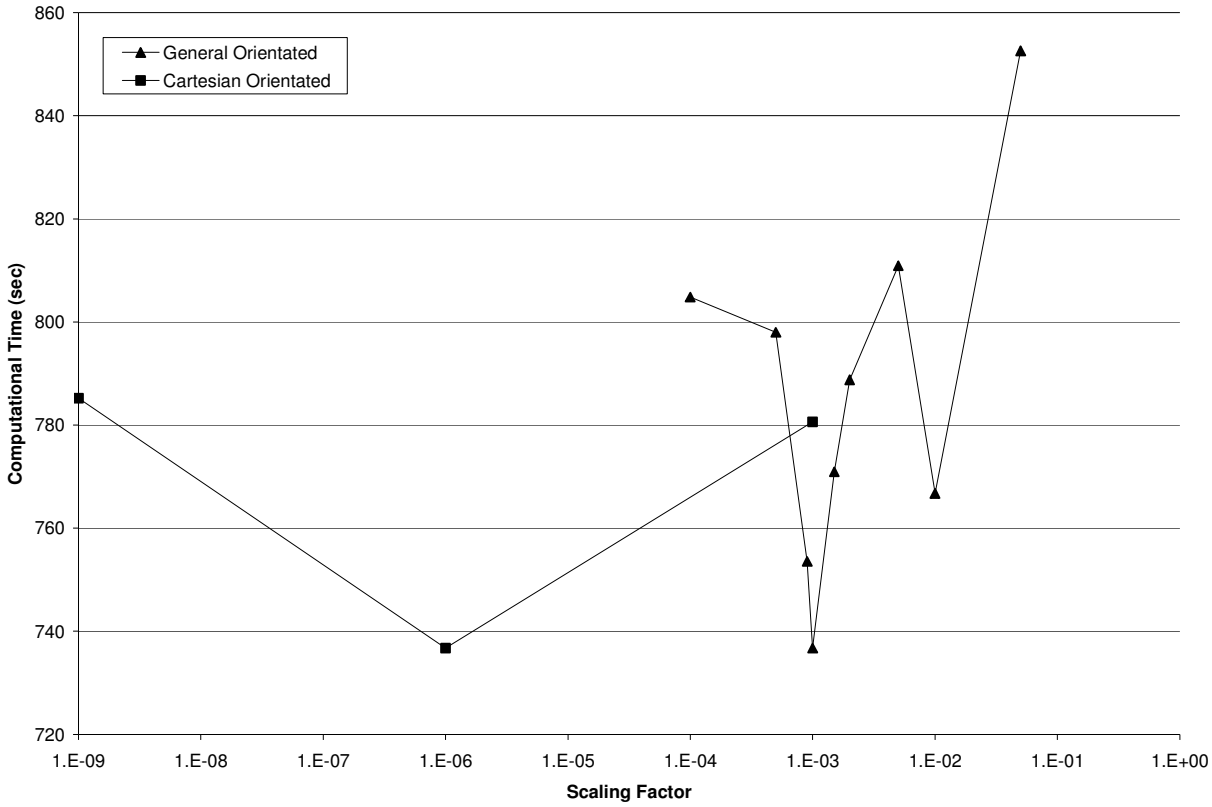


Figure XVII Timing Impact of the Scaling Factor

As can be seen, there is a 15% variation in the computational performance of the PN2ND solver depending upon the magnitude of the scaling factor. Such changes are expected since we are fundamentally changing the spectral radius of the coefficient matrix by introducing this factor. With additional calculations we were able to see that using a scaling factor > 0.1 led either to massively increased computational times or failure to converge within the specified iteration limits (i.e. the jobs were killed). This is also expected given the way the scaling factor impacts Eq. (A.7). Decreasing the scaling factor below 10^{-3} appears to increase the time, which we believe to be attributable to the poor

condition number of the coefficient matrix in Eq. (A.7). As a consequence, the settings of 10^{-3} for the generally orientated boundary condition combined with 10^{-6} on the Cartesian appear to be the best. While we have not currently linked the scaling factor to a group dependent quantity, we may later implement such an approach if we find that the current settings are found to be unreliable. Regardless of this, it is important to note that the eigenvalue using the new version is identical for all values of the scaling factor which was not the case for the old version of PN2ND. Consequently, the only concern of this scaling factor in the new version is its impact on the computational performance.

Appendix B

SN2ND Reflected Boundary Condition Treatment

B-1. SN2ND Reflected Boundary Condition Treatment

Unlike the spherical harmonics method, imposition of a reflected boundary condition in a second-order discrete ordinates discretization of the transport equation inherently requires that the cubature obeys all of the symmetry. All cubature that do not obey the symmetry will fundamentally make an approximation of some form to the reflected boundary condition operator which we have incorporated in the SN2ND solver via a least squares approximation. Similar to the PN2ND solver, for the finite element method, we impose these constraints on those spatial vertices that lie on the boundary of the reflected boundary condition. For those vertices that lie on apexes of multiple reflected boundary conditions we must impose all of the symmetry conditions which can at worst reduce the system of angular trial functions to a single free angular direction. To demonstrate the approach used in SN2ND, we use a simplified derivation starting with the system of equations

$$A_n \psi_n^+ = S_n^+ , \tag{B.1}$$

where ψ_n^+ is the even-parity flux for a given direction, A_n is the coefficient matrix for the within group even-parity flux for that direction and S_n^+ is the fixed source, fission source, and in scatter contribution for the current group and direction. In this system of equations we assume that all vacuum (or void) boundary conditions are accounted for and all that is left is the imposition of the reflected boundary conditions. For a reflected boundary surface, the reflected boundary condition imposes symmetry conditions on the set of angular directions and we can view this as a truncation of the existing set of trial functions that we represent as:

$$\begin{bmatrix} \psi_{ref,1}^+ \\ \psi_{ref,2}^+ \\ \vdots \\ \psi_{ref,N}^+ \end{bmatrix} = \tau_{ref} \begin{bmatrix} \psi_1^+ \\ \psi_2^+ \\ \vdots \\ \psi_N^+ \end{bmatrix} \rightarrow \psi_{ref}^+ = \tau_{ref} \psi^+ \tag{B.2}$$

where τ_{ref} is a square matrix with respect to the number of angular directions in the cubature. As an example, Eq. (B.3) shows the matrix for the four angular directions of the S_2 cubature (direction correspondence is provided above the τ_{ref} matrix definition) on a three-dimensional boundary surface with a (1,0,0) outward normal.

$$a = \frac{1}{\sqrt{3}} \quad \begin{array}{c|cccc} \mu & a & -a & -a & a \\ \eta & a & a & -a & -a \\ \zeta & a & a & a & a \end{array}$$

$$\tau_{(1,0,0)}^T = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{B.3}$$

It is important to note that the selection of the first and fourth directions is completely arbitrary and that we could have just as easily selected the second and third directions. We can implement Eq. (B.2) into Eq. (B.1) to obtain

$$\begin{aligned} \mathrm{T}_{ref} A_g \mathrm{T}_{ref}^T \mathrm{T}_{ref} \psi_g^+ &= \mathrm{T}_{ref} S_g^+ \\ A_{ref} \psi_{g,ref}^+ &= S_{g,ref}^+ \end{aligned} \tag{B.4}$$

where T_{ref} is a non square Boolean type matrix. For those vertices that are not along the reflected boundary condition, the corresponding angular component of T_{ref} is an identity matrix. For those vertices that are on the reflected boundary condition, the angular truncation matrix in Eq. (B.3) is used. We note that not only is the angular flux truncated to satisfy Eq. (B.2), but the set of trial functions that were used to weight the system of equations is also truncated, thus the presence of T_{ref} on both sides of the coefficient matrix in Eq. (B.4).

As was the case with the PN2ND solver, implementing this approach requires the definition of a space-angle connectivity list to handle the reduced set of trial functions that occurs. This causes problems with the parallelism of the solver and is fundamentally undesirable given that the source operation is substantially complicated. As a consequence, we formulated an alternative approach which obviates the need to truncate the system. To start we first define τ_{ref} in Eq. (B.3) to be square as shown in Eq. (B.5).

$$a = \frac{1}{\sqrt{3}} \quad \begin{array}{c|cccc} \mu & a & -a & -a & a \\ \eta & a & a & -a & -a \\ \zeta & a & a & a & a \end{array}$$

$$\tau_{(1,0,0)}^T = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{B.5}$$

We then define the angular matrix π_{ref} as

$$I - \tau_{ref}^T \tau_{ref} = \pi_{ref} \,. \tag{B.6}$$

We can add this onto both sides of a modified Eq. (B.4) and obtain a form similar to that of the PN2ND solver in Appendix A to get

$$\begin{aligned} \left(\mathrm{T}_{ref} A \, \mathrm{T}_{ref}^T + f \cdot \Pi_{ref}\right) \psi_{ref}^+ &= \mathrm{T}_{ref} S^+ + \left(f \cdot \Pi_{ref}\right) \psi_{ref}^+ \\ A_{ref} \psi_{ref}^+ &= \mathrm{T}_{ref} S^+ + \left(f \cdot \Pi_{ref}\right) \psi_{ref}^+ \end{aligned}, \tag{B.7}$$

The matrix Π_{ref} only contains π_{ref} for those vertices that lie on the reflected boundary conditions. The number f is an arbitrary scaling factor that has two purposes:

- Prevent the coefficient matrix in Eq. (B.7) from being singular
- Produce the best possible condition number

Because the SN2ND solver will always explicitly separate the set of directions into independent (non-zero rows), and dependent (zeroed rows), we can set the scaling factor to zero in Eq. (B.7) since the "zeroed" angular directions are no longer connected. This allows us to implement

$$A_{ref}\,\psi^{+}_{ref} = \mathrm{T}_{ref}\,S^{+}, \tag{B.8}$$

in a matrix free approach where only operations of $\mathrm{T}_{ref}\,A_g\,\mathrm{T}^{T}_{ref}\,\psi^{+}_{ref}$ occur. During the solution operation we simply apply T_{ref} to the discrete ordinate source for each within group equation and thereby correctly define the source for the reduced system as seen in Eq. (B.8). We note that this operation must be imbedded in the scattering source iteration that is generally applied in discrete ordinates methods.

Nuclear Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439-4842

www.anl.gov

A U.S. Department of Energy laboratory managed by UChicago Argonne, LLC